



Blender wiki PDF Manual conversion by Marco Ardito

Details, info, download: <http://amrc.altervista.org>

Updated: 09/10/2014 from:

<http://wiki.blender.org/index.php/Doc:CZ/2.6/Manual>





## Table of Content

Table of Content	2
Co je Blender?	16
Nebojte se Blenderu	16
Video: From Blender 1.60 to 2.50	18
Verze/Významné události	18
IRC chat channels	21
Who uses Blender?	21
Audience	22
Learning CG and Blender	22
summary	24
Co nového (october 2013)	24
Česká verze	24
Stažení a instalace binárky	25
Hardwarová podpora	25
Vývojové platformy	25
Kompilace zdrojáku	26
Kompilace pluginů	26
Průběžně ukládejte svou práci	29
Pojetí uživatelského rozhraní	29
Tři pravidla	29
Výkonné rozhraní	29
Přehled	29
Zápisy používané v tomto manuálu	31
Ovládání	31
Emulace tlačítek myši	31
Emulace NumPadu	31
Výchozí scéna	32
Informační okno (Info Window) (též hlavní horní menu)	32
3D Window View	33
3D Window Header	33
Buttons (Properties) Window Header	33
Outliner	33
Timeline Window ('Okno časové osy')	33
Maximalizace rámu	34
Dělení rámu	34
Spojování rámu	34
Změna velikosti rámu	34
Záměna obsahu rámu	34
Otevírání nových oken	34
Skryté záhlaví	35
Odkryté záhlaví	35
Pozice záhlaví	35
Tlačítko typu okna	35
Menu a tlačítka	35
Windows XP/Vista/7	36
Linux	36
MacOS	36
Stavové a chybové hlášky v okně konzole	37
Běžné zprávy	37
Nastavení Obrazovek	39
Přidání nové obrazovky	39
Smazání obrazovky	39
Upravení Obrazovky	39
Overriding Defaults	39
Additional Layouts	39
Konfigurace scén	41
Přidání nové scény	41
Stručný příklad	41
Odstranění scény	42
Cesty k souborům	43
Cesty ke skriptům	43
Uložit & Načíst	43

utomatické ukládání	44
Popis	46
Použití správce souborů a navigace v adresářích	46
Boční panel	46
Systém	46
Záložky	46
Poslední - Recent	46
Volby otevření - Open Options	46
Panel hlavičky	47
Další volby otevírání	47
Bezpečnost	47
Ochrana	47
Popis	49
Možnosti	49
Zobrazení vrstev	49
Přesouvání objektů mezi vrstvami	49
Animace vrstev	49
Příklad uspořádání vrstev	49
Skript pojmenovávající vrstvy	50
Typy objektů	52
Středy objektu	52
Posuny středu objektu	52
Vymazání objektu	53
Spojení objektů	53
Výběry a aktivní objekt	54
Výběr bodů	54
Výběr obdélníkem	54
Popis	54
Příklad	54
Tipy	55
Výběr lasem	55
Popis	55
Použití	55
Výběr kruhem	55
Popis	55
Výběr pomocí menu	55
Výběr seskupených	56
Popis	56
Volby	56
Výběr linkovaných	56
Popis	56
Volby	57
Výběr všech podle typu	57
Popis	57
Volby	57
Výběr všech podle vrstvy	57
Popis	57
Volby	58
Další volby menu	58
Popis	61
Příklady	61
Linked Duplicates	61
Description	61
Examples	61
Procedural Duplication	62
Linked Library Duplication	62
Hints	62
Duplikace vrcholů jako aranžovací nástroj	63
Nastavení	63
Přearanžování	63
Orientace	63
Duplikace vrcholů jako modelovací nástroj	64
Viz také	64
Základní použití	65
Škálování	65
Základní použití	66
Duplikace skupin a dynamické vazby	66
Příklad	66

Vytvoření duplikovaného objektu do reality	66
Příklady	67
<b>Externí odkazy</b>	<b>68</b>
<b>Sledování</b>	<b>69</b>
Popis	69
<b>Typy sledování</b>	<b>69</b>
Sledování To Constraint	69
Locked Track Constraint	69
Tlumené sledování	69
Staré sledování	69
<b>Hints</b>	<b>70</b>
Invalid Tracking or settings	70
Vstup do editačního režimu	71
Vizualizace	71
Tool Shelf	72
Properties Shelf	72
Zobrazení sítě (mesh)	72
Vrcholy	73
Hrany	73
Stěny	73
<b>Smyčky (loops)</b>	<b>73</b>
Smyčky hran	73
Smyčky ploch	74
Rovina - Plane	75
Krychle	75
Kruh	75
Koule - UV Sphere	76
Koule - Icosphere	76
Válec	76
Kužel	76
Anuloid - torus (pneumatika)	76
Mříž	77
Opička	77
Doplňky	77
Overhang	78
Thickness	78
Intersections	78
Distortion	78
Sharp Edges	78
<b>Režimy výběru</b>	<b>80</b>
Select Mode Header Widgets	80
Select Mode Pop-up	80
Switching select mode	80
<b>Nástroje výběru</b>	<b>81</b>
<b>Režimy výběru</b>	<b>83</b>
Select Mode popup	83
Select Mode header widgets	83
Výběr prvků po přepnutí režimu výběru	83
Přidávání do výběru	85
Výběr prvků v regionu	85
Obdélníková oblast (výběr pozadí)	85
Kruhový region	85
Region laso	86
Doplňkové nástroje výběru	86
Výběr podobných	87
Vybírání smyček	88
Smyčky hran a vrcholů	88
Face Loops	88
Edge Ring	89
Path Selection	89
Loop Inner-Region	90
Boundary Loop	90
Smyčky hran	92
Prstence hran	92
Smyčky stěn	94
N-úhelníky v režimu výběru stěn	94
<b>Typy nástrojů</b>	<b>95</b>
<b>Přístup k nástrojům sítě</b>	<b>96</b>
Paleta nástrojů	96

Menu	96
Smazat (Delete)	100
Konverze trojúhelníků na čtyřúhelníky	100
Propojení, slítí (Unsubdivide)	100
Spojování	100
Spojování vrcholů	100
AutoMerge Editing	101
Remove Doubles	101
Další odkazy	103
Zrcadlení-X	104
Modifikátor zrcadla	104
Symmetrizace sítě	104
Zrcadlení geometrie	104
<b>Spojování (Merging)</b>	<b>105</b>
Spojování vrcholů	105
Automatické spojování	105
Odstranění duplicit	105
<b>Oddělení</b>	<b>105</b>
Roztřížení	105
Příklady	106
Omezení	107
Vyplněné roztřížení	107
Split	108
Oddělení	108
<b>Propojení vrcholů</b>	<b>108</b>
<b>Vertex Slide</b>	<b>109</b>
<b>Vyhlazení (Smooth)</b>	<b>109</b>
<b>Vytvoření z vrcholu rodiče</b>	<b>109</b>
<b>Přidání háku (Hook)</b>	<b>109</b>
<b>Mísení ze tvarů, množení tvarů</b>	<b>110</b>
<b>Vytvoření hrany/stěny</b>	<b>111</b>
<b>Nastavení atributů hran</b>	<b>111</b>
Mark Seam and Clear Seam	111
Mark Sharp and Clear Sharp	111
Adjust Bevel Weight	111
Crease SubSurf	111
<b>Posuv hrany</b>	<b>111</b>
Použití	112
Even mode	112
Omezení & Workarounds	113
<b>Rotace hrany</b>	<b>113</b>
Použití výběru stěn	113
<b>Smazání smyčky hran</b>	<b>114</b>
Limitations & Workarounds	114
Example	114
<b>Collapse</b>	<b>114</b>
<b>Edge Split</b>	<b>115</b>
<b>Vytváření stěnes</b>	<b>116</b>
Vytvoření hrany/stěny	116
Vyplnění	116
Krásné vyplnění	116
Konverze čtyřúhelníků na trojúhelníky	117
Konverze trojúhelníků na čtyřúhelníky	117
<b>Solidify</b>	<b>118</b>
<b>Rotace hran</b>	<b>119</b>
<b>Normály</b>	<b>119</b>
Překlopení směru	119
Přepočítání normál	119
Vytlačení/vytažení (Push/Pull)	121
Pokřivení (Warp)	121
Příklad	121
Smýkání, zkosení (Shear)	122
Do koule (To Sphere)	123
Příklad	123
Osy symetrie (symetrály)	125

Pivot point	125
Transformation orientation	126
Laplaceovo uhlazení	128
Vytáhnout region	132
Individuální vytažení	132
Vytažení pouze hran a vrcholů	132
Volby	135
<b>Příklady</b>	<b>136</b>
Úhel	137
Dupli	137
Spojit duplikáty	138
Přepočítání normál	138
Volby	141
<b>Příklady</b>	<b>142</b>
Jedna hrana	142
Dva trojúhelníky	143
Dvě protilehlé hrany čtyřúhelníku	143
Dvě sousedící hrany čtyřúhelníku	144
Tři hrany	145
Režim "Tri mode"	145
Čtyřúhelník/čtyři hrany	146
Vícenásobné dělení	146
Využití	148
Volby	148
Usage	150
Options	150
Confirming and selection	152
Limitations	152
Optimizations	152
<b>Knife Project</b>	<b>152</b>
Examples	152
Known Issues	153
Postup	154
Modifikátor zkosení	157
Použití	157
Volby, nastavení	157
Příklady	158
Seřadit Prvky (Elements)	159
Přilepování k sítí	160
Modifikátor Shrinkwrap	160
<b>Režim váhového kreslení</b>	<b>162</b>
Barevné stupně váhy	162
<b>Štětce</b>	<b>162</b>
Vlastnosti hlavního štětce	163
Volby normalizace	163
Definice tvaru čáry	163
Křivka přitlaku štětce	164
Projevy štětce	164
Přednastavené štětce	164
Přednastavení barevného prostoru kreslení	165
<b>Selection Masking</b>	<b>165</b>
Details about selecting	165
Vertex Selection Masking	165
Face Selection Masking	166
Hide/Unhide Faces	166
Hide/Unhide Vertices	166
The Clipping Border	166
<b>Weight Paint Options</b>	<b>166</b>
<b>Weight Paint Tools</b>	<b>167</b>
<b>Weight Painting for Bones</b>	<b>167</b>
<b>Weight Painting for Particles</b>	<b>168</b>
Volba podmnožiny (The Subset Option)	169
<b>Normalize All</b>	<b>169</b>
<b>Normalizace všech</b>	<b>169</b>
Parametry operátoru	169
<b>Normalizace (Normalize)</b>	<b>169</b>

Parametry operátoru	169
<b>Zrcadlení (Mirror)</b>	<b>169</b>
Parametry operátoru	170
<b>Inverze (Invert)</b>	<b>170</b>
Parametry operátoru	170
<b>Vyčištění (Clean)</b>	<b>171</b>
Parametry operátoru	171
<b>Úrovně (Levels)</b>	<b>171</b>
Parametry operátoru	171
<b>Promíchání (Blend)</b>	<b>172</b>
Parametry operátoru	172
<b>Přenos vah (Transfer Weights)</b>	<b>173</b>
Příprava kopie (Prepare the copy)	173
Spuštění nástroje (Call the tool)	173
Redo Panel Confusion	173
Obezička (Workaround)	173
Parametry operátoru	173
<b>Omezení počtu (Limit total)</b>	<b>174</b>
Parametry operátoru	174
<b>Gradient váhy (Weight Gradient wip)</b>	<b>174</b>
Parametry operátoru	174
<b>Mesh Shading</b>	<b>176</b>
Smooth shading	176
Smoothing parts of a mesh	176
Auto Smooth	176
Edge Split Modifier	177
Smoothing the mesh geometry	177
Mesh editing tools	177
Modifiers	177
Vyplnění děr	179
Rozdělení (Split) nerovinných stěn	179
Úvody	180
Křivková primitiva	180
Beziérové křivky	180
Úprava Beziérových křivek	180
Vlastnosti křivek	181
Tvar	181
Geometry	182
Path Animation	183
Active Spline	183
Non-Uniform Rational B-Splines (NURBS)	184
Editing NURBS Curves	184
Active Spline	184
Path	184
Menu pro výběr	186
Každý n-tý	186
Vybrat/odvybrat první/poslední	186
Vybrat další/předchozí	186
Více a méně	186
Vytvoření křivek	188
Zobrazení křivek	188
Možnost i zobrazení	188
Skryté prvky	188
Základní úpravy křivek (posuv, rotace, měřítko)	188
Přichytávání	188
Nástroje deformace	189
Vyhlažování	189
Zrcadlení	189
Set Bézier Handle Type	189
Rozšiřování křivek	189
Podrozdělení	189
Duplikace	190
Spojování segmentů křivek	190
Oddělení křivek	190
Vymazání prvků	190
Otevírání a uzavírání křivek	190
Přepnutí směru	191
Nástroje pro konverzi	191
Typy převodu křivky	191

Převod křivky na síť	191
Převod sítě na křivku	191
Rodičovství křivek	191
Háky	191
Nastavení cílové váhy	192
<b>Nástroje úpravy povrchů</b>	<b>193</b>
<b>Vizualizace</b>	<b>193</b>
<b>Struktura povrchu</b>	<b>193</b>
Řídící body, řádky a mřížka (Grid)	193
Rozlišení povrchu	194
Uzavřené a otevřené povrchy	194
Uzly (Knots)	194
Pořadí (Order)	194
Váhy (Weight)	195
Přednastavené váhy	195
Primitiva	195
Notes	197
<b>Editing Text</b>	<b>197</b>
Inserting Text	198
Special Characters	198
Convert text to text object	198
3D Mesh	198
Text Selection	199
<b>Formatting Text</b>	<b>199</b>
Fonts	199
Loading and Changing Fonts	199
Size and Shear	200
Objects as Fonts	200
Text on Curve	200
Underline	200
Character	201
Setting Case	201
Paragraph	201
Align	201
Spacing	202
Offset	202
<b>Shape</b>	<b>202</b>
Resolution	202
Fill	202
Textures	202
<b>Geometry</b>	<b>203</b>
Textové boxy	204
Velikost rámu	204
Přidání/mazání rámu	204
Příklad: Tok textu	204
Příklad: Více sloupců	205
Přiřazení materiálu	205
Primitiva	206
<b>Vizualizace</b>	<b>206</b>
<b>Volby pro Meta Ball</b>	<b>206</b>
Rozlišení	207
Threshold (Influence)	207
Update	207
<b>Meta Structure</b>	<b>208</b>
Technical Details	208
Underlying Structure	208
<b>Nastavení</b>	<b>210</b>
<b>Použití a funkce</b>	<b>210</b>
Vizualizace	212
Úprava (Modify)	213
Vytváření (Generate)	213
Deformace (Deform)	214
Simulace	214
<b>Rozhraní (Interface)</b>	<b>215</b>
<b>Svazek (Stack)</b>	<b>215</b>
Volby	217
Šířka	217

Segmenty	217
Pouze vrcholy (Only Vertices)	217
Metody omezení (Limit Method)	218
Popis	219
Volby	219
Příklady	219
Jednoduchá rovina	219
Zjednodušený válec	219
Popis	221
Volby	221
Hints	222
Použití modifikátoru zrcadlení spolu s modifikátorem podrozdělení povrchu (Subdivision Surface)	222
Zarovnávání se zrcadlem	222
Popis	223
Volby	223
Volby	224
Příklad	224
<b>Externí odkazy</b>	<b>225</b>
Popis	226
Nastavení	226
Popis	227
Volby	227
<b>Nastavení</b>	<b>228</b>
Interakce Soft Body a Oblečení	229
Interakce se silovými poli	229
Příklady	229
Tipy	229
<b>Postup - Workflow</b>	<b>230</b>
<b>Vytvoření částicového systému</b>	<b>231</b>
Typy částicových systémů	231
Společné volby	231
<b>Odkazy</b>	<b>231</b>
Omezení a vlivy pohledu	232
Globální vlivy	232
Nastavení osvětlení	232
Lighting in the Workflow	232
Overriding Materials to Reset Lighting	232
Jak materiály fungují	234
Použití materiálů	235
Materiálové textury	236
World Textures	237
Brush Textures	237
Popis	239
Volby	239
Armatury	240
Editování	240
Skinning-povrchování	240
Posing - pozicování	240
Vaše první armatura	242
Objekt armatura	242
Přehled kapitoly o armaturách	242
Bones Visualization	244
Bones properties	245
Bones Rigidity	245
Bones influence	245
<b>Panel zobrazení</b>	<b>247</b>
Typy kostí	247
Kost osmistěn	247
Kost tyčka	248
Kost B-Bone	248
Kost obálka	248
Atributy	248
Tvary kostí	249
= Atributy	249
<b>Armature Layers</b>	<b>250</b>
Showing/hiding bone layers	250
Protected Layers	250
<b>Vrstvy kostí</b>	<b>250</b>

Přesun kostí mezi vsrtvami	250
<b>Skrývání kostí</b>	<b>250</b>
Vizualizace kostí	252
Vlastnosti kosti	253
Bones Rigidity	253
Bones influence	253
Transformování kostí	255
Radius and Scaling in Envelope Visualization	255
ScaleB a Envelope - obálka	255
Směr kosti - Bone Direction	256
Bone Roll	256
Properties	257
Bone Rigidity Settings	257
Chain Editing	258
Naming Bones	259
Naming Conventions	259
Bone name flipping	260
Auto bone naming	260
Úvod	261
Panel knihovny pozic	261
Úpravy	261
Použití omezovačů	262
Omezování kostí	262
Dostupné omezovače	262
Sledování pohybu	262
Transformační omezovače	262
Dráhové omezovače	263
Vztahové omezovače	263
Záhlaví omezovače	264
Nastavení omezovače	264
Cíl (target)	264
= Prostor (Space) omezovače	265
Vliv (Influence)	265
Přidání/ostranění omezovače	266
Popis	267
Volby	267
Volby	268
Panel klíčových sad	271
Panel aktivní klíčové sady	271
Přidávání vlastností	273
Viz také	273
<b>Prvky časové osy</b>	<b>274</b>
Aktuální snímek	274
Klíčové snímky	274
Značky (Markers)	274
<b>Nastavení pohledu</b>	<b>274</b>
Menu pohled	275
<b>Úpravy</b>	<b>275</b>
Menu rámečků	275
<b>Přehrát</b>	<b>275</b>
Menu přehrávání	275
Header Controls	275
<b>Vytváření klíčových snímků</b>	<b>276</b>
v pohledu 3D	276
v panelu vlastností	276
v animačním editoru	277
<b>Úpravy klíčových snímků</b>	<b>277</b>
Gravitace	278
Vytvoření silového pole	279
<b>Common Field Settings</b>	<b>279</b>
Falloff	279
<b>Typy polí</b>	<b>280</b>
silové	280
vítr	280
vírové	280
magnetické	280
harmonické	280

náboj	280
Lennard-Jonesovo	281
Pole textur	281
Příklady	281
Curve Guide	281
Boid	282
Turbulence	283
Tažení, brždění	283
Odkazy	283
<b>Nastavení</b>	<b>284</b>
Interakce Soft Body a Oblečení	285
Interakce se silovými poli	285
Příklady	285
Tipy	285
<b>Postup - Workflow</b>	<b>286</b>
<b>Vytvoření částicového systému</b>	<b>287</b>
Typy částicových systémů	287
Společné volby	287
<b>Odkazy</b>	<b>287</b>
Úvod	288
Začínáme	288
Dohlížení na 2D tracking	288
Manuální kalibrace objektivu pomocí grease pencil a/nebo mřížky	288
Řešení pohybu kamery	288
Základní nástroje pro orientaci a stabilizaci scény	288
Základní uzly pro zakomponování scény do reálného záznamu	288
Neimplementované nástroje	288
Manuál	288
Movie Clip Editor	288
Tools available in tracking mode	290
Markers panel	290
Track panel	290
Solve panel	290
Cleanup Panel	290
Clip Panel	291
Properties available in tracking mode	291
Grease Pencil Panel	291
Track Panel	291
Camera Data Panel	291
Display Panel	292
Tracking Settings Panel	292
Common options	292
KLT tracker options	293
SAD tracker options	293
Marker Panel	293
Proxy/ Timecode Panel	293
Tools available in reconstruction mode	293
<b>Proces renderování</b>	<b>295</b>
Přehled	295
Distribuovaná renderovací farma	295
Integrace pracovního Renderu Workbench	295
<b>Panel nastavení renderu</b>	<b>296</b>
Render	296
Vsrty	296
Rozměry Dimensions	296
Anti-Aliasing	296
Motion Blur -	296
Shading - stínování	297
Výstup	297
Výkon	297
Post Processing	297
Razítka, označení	297
Bake (zapečení)	297
<b>Přidání nové kamery</b>	<b>298</b>
<b>Změna aktivní kamery</b>	<b>298</b>
<b>Nastavení kamery</b>	<b>298</b>
Čočky	298
Display	299
Composition Guides	300
<b>Camera Navigation</b>	<b>300</b>
Move active camera to view	300
Camera View Positioning	300
Roll, Pan, Dolly, and Track	300
Aiming the camera in Flymode	301

<b>Syntaxe</b>	<b>302</b>
Volby Renderu:	302
Volby formátu:	302
Volby playbacku:	302
Volby okna: Options:	303
Specifické volby pro herní engine:	303
Volby prostředí Python:	303
Volby debugování:	303
Různé volby:	304
Další volby:	304
<b>Příklady</b>	<b>304</b>
Renderování obrázku	304
Renderování filmu	304
Spuštění Blenderu se specifickým engine	305
<b>Platformy</b>	<b>305</b>
Windows	305
Mac OS X	305
Linux	305
Začínáme	307
Odkazy	307
Renderování pomocí GPU	308
Úvody	308
Perspektiva	311
Ortografické zobrazení	311
Panoramatické zobrazení	311
Equirectangular	311
Rybí oko	311
Hloubka ostroty (Depth of Field)	311
Clipping	312
Shader povrchu	313
Volume Shader	313
Zdvihový shader (Displacement)	313
Úspora energie	313
Typy výstupků	314
Podrozdělení - Subdivision	314
Terminologie	315
Parametry BSDF	315
Volume Shaders	316
Density	316
Volume Material	316
Interaction with the Surface Shader	316
Mesh Topology	316
Volume World	316
Scattering Bounces	317
Omezení	317
Hlasitosti Shadery	317
Objem Materiál	317
Interakci s povrchem Shader	317
Mesh topologie	317
Objem World	317
Rozptyl skáče	318
<b>Svítidla (lamps / lamps)</b>	<b>319</b>
Bodové světlo	319
Spot Lamp	319
Plošné světlo (Area Lamp)	319
Sluneční (Sun) světlo	319
Shadery	320
Textury	320
Další	320
Open Shading Language	320
Kamera	321
Hodnota	321
RGB	321
Atribut	321
Geometrie	321
Cesta paprsku	321
Object Info	322
Fresnel	322
Layer Weight	322
Hmotnost vrstvy (Layer Weight)	323

Souřadnice textur	323
Informace o částicích	323
Informace o vlasech	324
Tangentá (tečna)	324
Výstup z materiálu	325
Výstup z lampy	325
Výstup světový (World)	325
Typy záření	326
Ovládání odrazů	326
Průsvitnost (Transparency)	326
Viditelnost	327
Základní kontext	329
Známa omezení	329
Pokročilé možnosti	331
Výběr podle Viditelnosti	333
Výběr podle typů hran	334
Výběr podle typy hran	334
Edge Marks	335
Selection by Face Marks	335
Selection by Group	336
Selection by Image Border	337
Řetězení - Chaining	339
Půlení - Splitting	339
Výběr	339
Zakončení	340
Čerchovaná čára - Dashed Line	340
Modifikátory	341
Podélné tahy	341
Vzdálenost od kamery	341
Vzdálenost od objektu	341
Materiál	342
Modifikátory	343
Podélné tahy	343
Vzdálenost od kamery - Distance from Camera	343
Vzdálenost od objektu - Distance from Object	344
Materiál	344
Modifikátory	345
Podélné tahy	345
Kaligrafie	345
Vzdálenost od kamery	346
Vzdálenost od objektu	346
Materiál	347
Modifikátory	348
2D Posuv	348
Transformace 2D	348
Backbone Stretcher	349
Bezier Curve	349
Blueprint	349
Guiding Lines	350
Perlin Noise 1D	350
Perlin Noise 2D	350
Polygonization	351
Sampling	351
Sinus Displacement	351
Spatial Noise	352
Tip Remover	352
Vytvoření modulu pro styl	353
Výběr	354
Chaining-řetězení	354
Rozdělení	354
Trídění	354
Vytvoření tahů	355
User control on the pipeline definition	355
Videa	356
Video úvodníky	356
Úvody	356
Různé	356
Přístup a aktivace uzlů	357
Příklady	358
Addons	361
Skripty	361
Ukládání vlastních skriptů	361
Umístění souborů	361

Instalace	361
-----------	-----



## Úvod



Blender 2.5 s Big Buck Bunnym

Vítejte v uživatelské dokumentaci Blenderu! Dokumentace Blenderu sestává z mnoha částí: tento uživatelský manuál, referenční příručka, návody, fora, a spousta dalších webových zdrojů. První část tohoto manuálu vám ukáže jak Blender stáhnout a nainstalovat. Pokud si zvolíte stažení zdrojového kódu, pak se také dozvíte jak Blender zkompileovat.

Blender má uživatelské rozhraní, který je optimalizované pro tvorbu 3D grafiky. Z počátku může být trochu matoucí pro nové uživatele, ale v uživatelském rozhraní Blenderu je jeho největší síla. Sekce [Rozhraní](#) vás blíže seznámí s rozhraním a jeho konvencemi.

## Co je Blender?

Blenderu vznikl již v roce 1994 jako integrovaná aplikace, která umožňuje vytváření 2D a 3D obsahu. Blender poskytuje široké spektrum nástrojů pro modelování, texturování, osvětlení, animace a video post-processingu v jednom uceleném softwarovém balíku. Díky své otevřené architektuře poskytuje multiplatformnost, rozšiřitelnost a pevně ucelený workflow. Blender je jednou z nejpoužívanějších 3D grafických aplikací na světě.

Blender může být použit k tvorbě 3D vizualizací, stejně tak i pro kvalitní video, nicméně je v něm včleněn i real-time 3d engine sloužící k tvorbě interaktivního obsahu a her.

Blender byl původně vyvíjen společností 'Not a Number' (NaN). Nyní je Blender 'Free Software' a jeho zdrojový kód je šířen pod licencí GNU GPL. Jeho vývoj probíhá pod záštitou a za finanční podpory Blender Foundation.

Mezi lety 2008 a 2010 byl Blender kompletně přepracován, jeho funkce, rozhraní a workflow byly vylepšeny. Výsledkem této snahy je verze softwaru známá jako Blender 2.5.

Hlavní rysy:



Renderování obrázku a jeho postprodukce

- Plně integrovaná sada nástrojů nezbytných pro 3D tvorbu, jako je: modelování, UV mapování, texturování, animování, částice, simulace, skriptování, renderování, post-produkci videa a tvorbu her
- Multiplatformní aplikace s rozhraním založeným na OpenGL, kterou lze provozovat na: Windows, Linux, OSX, Solaris a mnoho dalších.
- Umožňuje rychle a produktivně vytvářet velmi kvalitní 3D architekturu
- Podpora uživatelské komunity: na <http://BlenderArtists.org>, novinky: <http://BlenderNation.com> a česká podpora na <http://www.blender3d.cz>
- Malý spustitelný soubor pro snadnější distribuci

Aktuální verzi Blenderu si můžete stáhnout [zde](#).

## Nebojte se Blenderu



Uvítací obrazovka podzimu 2013

Přestože plnohodnotný systém Blender obsahuje tisíce různých funkcí, od úpravy grafiky, modelování, fyzikálních simulací, videoprocessingu, renderingu... nemusíte mít strach, že Blender "nezvládnete"! Systém Blender má totiž gigantickou výhodu: stačí porozumět několika nástrojům uvnitř a hned dokážete vytvářet svá díla bez nutnosti znalosti či zkušeností nad detailními oblastmi (například kinematiky). Stačí vzít, špetku odvahy a začít.

Ovládání systému se může pro ty, kteří s podobnými systémy ještě nepracovali zdát poněkud komplikovaný. Jsou využity všechna tři tlačítka myši, je zde velké množství horkých kláves... Ale jak zaznělo výše: veškeré funkce jsou dostupné z menu a stačí pouze několik dní hraní si s Blenderem a máte jeho základní principy "pod kůží".

Po startu Blenderu jste uvítáni obrazovkou, jejíž nosná fotografie je samozřejmě též vytvořena v Blenderu. Každá nová verze má novou fotografii. Může sloužit jako motivace pro grafika, který právě startuje blender.

## Historie Blenderu

V roce 1988 založil Ton Roosendaal v Holandsku animační studio NeoGeo. Brzy se z něj stalo největší animační studio v Holandsku a jedno z předních v Evropě. NeoGeo vytvářelo významné oceněné produkty (European Corporate Video Awards 1993 a 1995) pro mnoho velkých korporací, například Philips. V NeoGeo byl Ton mimo jiné zodpovědný za výtvarný směr a vývoj softwaru. Po pečlivé úvaze se Ton rozhodl, že aktuálně používaná sada 3D nástrojů je v NeoGeo zastaralá a těžkopádná pro vylepšení a že je třeba ji od základu přepsat. Práce započaly v roce 1995 a brzy spatřil světlo světa program, který je dnes znám jako Blender. V době, kdy NeoGeo dokončovalo úpravy na vylepšení Blenderu, Ton přišel s myšlenkou, že Blender by mohli používat i další umělci, nejen zaměstnanci NeoGeo.

V roce 1998 se Ton rozhodl založit další firmu pod názvem „Not a Number“ (NaN) jako odnož NeoGeo, se zaměřením na větší trh a zabývající se rozvojem Blenderu. Ideou NaN bylo vytvářet a distribuovat zdarma malý, multiplatformní program pro tvorbu 3D grafiky. V té době to byla revoluční myšlenka, protože cena většiny komerčních programů pro 3D grafiku byla několik tisíc dolarů. NaN doufal, že vytvoří profesionální nástroj pro modelování a animaci, který by byl dostupný pro každého. Byznys model firmy NaN sázel na komerční produkty a služby kolem Blenderu. V roce 1999 se firma NaN poprvé zúčastnila konference Siggraph se snahou o větší propagaci Blenderu. Blender, který byl na Siggraphu poprvé, měl velký úspěch a velký zájem projevil nejen tisk, ale i návštěvníci. Blender byl hitem, což jen potvrdilo jeho potenciál.

Na křídlech úspěchu po Siggraphu na počátku roku 2000 byla NaN finančně zabezpečena částkou 4,5 milionu €, kterou získala od investorů. Velký přísun peněz umožnil rychlý růst společnosti a zkrátko se NaN mohla pochlubit více jak 50 pracovníky z celého světa, kteří pracovali na zlepšení a podpoře Blenderu. V létě 2000 byl vydán Blender 2.0. Tato verze přidala integrované herní rozhraní k 3D nástrojům. Do konce roku 2000 bylo na internetové stránce NaN zaregistrováno přes 250 000 uživatelů.

Bohužel ani ambice a dovednosti NaN nestačily k tomu, aby se firma uchytila na tehdejšímu trhu. Důsledkem toho bylo obnovení činnosti NaN s novým investorem a malou společností v květnu 2001. O šest měsíců později se na trhu objevil první komerční produkt NaN s názvem Blender Publisher. tento produkt byl zaměřen na nově vzniklý trh s interaktivními 3D medii pro WWW stránky. Bohužel zklamání z prodeje a těžká ekonomická situace vedla nové investory k tomu, že se rozhodli zastavit všechny práce NaN. Ukončení prací zastavilo také další rozvoj Blenderu. Mimo toho, že v aktuální verzi Blenderu byly velké nedostatky, zároveň se složeným vnitřním jádrem, málo výkonnými nástroji a nestandardním rozhraním, setkala se tato verze s nadšenou podporou uživatelské komunity, jejíž členové v minulosti Blender Publisher koupili. Ton nenašel důvody pro opuštění Blenderu, ale obnovit činnost firmy s malou skupinou pracovníků bylo nemožné. Proto v březnu 2002 založil nekomerční organizaci Blender Foundation.

Základním cílem Blender Foundation bylo najít způsob jak pokračovat v rozvoji a podpoře Blenderu jako [Open Source](#) projektu. V červenci 2002 se Tonovi podařilo přesvědčit investory NaN, aby souhlasili s jedinečným plánem uvolnit Blender jako open source. Kampaň „Free Blender“ („Volný Blender“) potřebovala získat 100,000 €, aby Blender Foundation mohla od investorů NaN koupit práva na zdrojový kód Blenderu a následně jej uvolnit pod Open Source licenci. Zároveň nadšená skupina dobrovolníků, z nich část byli bývalí zaměstnanci NaN, založila sbírku pro „Free Blender“. K údivu všech, kampaň získala potřebných 100,000 € během krátkých 7 týdnů. V neděli 13 října 2002 byl Blender vydán pod licenci [GNU General Public License \(GPL\)](#). Rozvoj Blenderu trvá dodnes, poháněný kolektivem mnoha oddaných dobrovolníků z celého světa, které řídí sám zakladatel Ton Roosendaal.

## Video: From Blender 1.60 to 2.50

[\[video link\]](#)

## Verze/Významné události

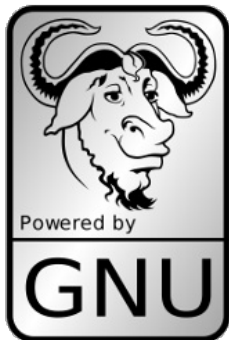
### Historie Blenderu a jeho vývoj

- 1.00 Leden 1995 Blender je vyvíjen v animačním studiu NeoGeo
- 1.23 Leden 1998 Představena SGI verze na webu, IrisGL
- 1.30 Duben 1998 Verze pro Linux a FreeBSD s výstupem do OpenGL a X
- 1.3x Červen 1998 Založena společnost NaN
- 1.4x Zář 1998 Uvolněna alfa verze pro systémy Sun a Linux
- 1.50 Listopad 1998 Publikován první manuál
- 1.60 Duben 1999 C-key (nový nástroj pro uzamčení, \$95), uvolněna verze pro Windows
- 1.6x Červen 1999 Uvolněna verze pro BeOS a PPC
- 1.80 Červen 2000 Ukončení C-key, Blender se opět stává freewarem
- 2.00 Srpen 2000 Interaktivní 3D a real-time engine
- 2.10 Prosinec 2000 Nový engine, fyzika, a Python
- 2.20 Srpen 2001 Character animation system
- 2.21 Říjen 2001 Vypuštěn Blender Publisher
- 2.2x Prosinec 2001 Mac OSX verze
- **13. Říjen 2002 Blender přechází na Open Source, první Blender Konference**
- 2.25 Říjen 2002 Blender Publisher volně ke stažení
- Tuhopuu1 Říjen 2002 Je vytvořena experimentální větev Blenderu pro volnou ruku vývojářům
- 2.26 Únor 2003 První opravdový Open Source Blender
- 2.27 Květen 2003 Druhý Open Source Blender
- 2.28x Červenec 2003 První ze série 2.28x
- 2.30 Říjen 2003 Druhá Blender Konference, představeno předělané UI verze 2.3x
- 2.31 Prosinec 2003 Zdokonalení stability UI projektu verze 2.3x
- 2.32 Leden 2004 Významné renovace schopností interního renderu
- 2.33 Duben 2004 Implementován Ambient Occlusion, nové procedurální textury a byl opět zakomponován game engine!
- 2.34 Srpen 2004 Velké vylepšení: Particle Interactions, LSCM UV mapping, funkční integrace YafRaye, Weighted creases v Subsurfaces, Ramp shadery, full OSA, a mnohem mnohem víc.
- 2.35 Listopad 2004 Další verze plná vylepšení: Object hooks, curve deforms a curve tapers, particle duplikátory atd.
- 2.36 Prosinec 2004 Stabilizační verze, pár nových věcí přidáno, jako jsou Normals a Displacement mapping
- 2.37 Červen 2005 Velký pokrok: Transformační nástroje, Softbodies, Force fields, deflections, incremental Subdivision

Surfaces, Transparentní stíny, a Multithreaded rendering.

- 2.40 Prosinec 2005 [An even bigger leap](#): plně přepracovaný systém armatur, shape keys, chlupy pomocí částic, kapaliny a rigid bodies.
- 2.41 Červen 2006 Převážně vylepšení Game enginu a [lots of fixes](#).
- 2.42 Červen 2006 Hlavním motivem jsou **Nody**. Přez [50 developers](#) přispělo při tvorbě Nodů, Array modifikátor, vector blur, nový engine fyziky, vylepšení renderingu, zlepšení synchronizace obrazu se zvukem a [many other features](#).
- 2.43 Únor 2007 Hlavním motivem je slovo **Multi**: Multi funkce: multi-resolution Meshes, multi-level UV Textures, multi-layer images a multi-pass rendering a baking, Sculpting, Retopology, multiple additional Matte, Distort a Filter Nody, vylepšení modelingu a animací, lepší kreslení s multiple brushy, fluid particles, proxy objects, přepsán byl Sequencer, a post-produkční UV texturing. Uff! Oh, a [website rewrite](#) A ano, je tu i možnost využívat multi-threaded rendering pro multi-core CPU. S Verze jako multi-user, můžete spolupracovat na stejném projektu ve více lidech ve stejném čase.
- [2.44 May 2007 SSS](#): Velké novinky, přidány dva nové modifikátory a oživení podpory 64-bit OS, byl přidán subsurface scattering, který simuluje rozptyl světla zpoza povrchu organických a měkkých objektů
- 2.45 Září 2007 Jedná se především o stabilizační verzi s opravou spousty chyb z předešlých verzí.
- 2.46 Květen 2008 [Peach release] byl výsledkem velkého úsilí více než 70 vývojářů, kteří vylepšili nebo pomáhali při vývoji hair and fur, nového particle systému, image browseru, cloth simulace, hladký a nerušivý physics cache, v renderingu byli vylepšeny reflexe, AO, a render baking; mesh deform modifier pro svalovinu a také lepší podpora pro animaci pomocí armature tools a drawing, skinning, constraints a barevnější Action Editor, a mnoho dalšího.
- 2.47 Srpen 2008 Bugfix release
- 2.48 Říjen 2008 [Apricot release](#): GLSL shadery, světla a GE vylepšení, Snap, sky simulátor, shrinkwrap modifikátor, vylepšená python editace
- 2.49 Červen 2009 [Pre-Re-Factor release](#) přidána významná zlepšení jádra a GE. Jádro bylo vylepšeno o texture node, armature sketching (nazván Etch-a-Ton), vylepšenými boolean, mesh operacemi, podporou JPEG2000, projection painting představující přímý přenos obrázku na modely, a Python script katalog. GE vylepšení obsahuje video textury, přes které můžete pouštět filmy ve hře (!), upgradován byl též i Bullet physics engine, dome (fish-eye) rendering, a více dostupných API GE volání.
- 2.5 – Od roku 2008 do září 2010. Tato řada [release](#) 4 před-verzí (od Alpha0 - Listopad 2009 - až po Beta Červenec 2010. Je to jeden z nejdůležitějších [projektů](#) blenderu s kompletním přepracováním - nové funkce, změna okenního systému a event/tool/data handling system, nové python API... Finální verze tohoto projektu bude Blender 2.6 (naplánováno na Červenec 2010).

O svobodném software a GPL



Když se někdo dozví o "svobodném softwaru", první věc, která na kterou myslí, je, že to je "zdarma". I když je to pravda, ve většině případů určuje termín "svobodný software", jak používat Free Software Foundation (původci projektu GNU a tvůrci GNU General Public License) znamená "svobodný ve smyslu svobody" spíše než ve smyslu "žádné náklady". Svobodný software v tomto smyslu je software, který můžete libovolně používat, kopírovat, modifikovat, distribuovat, bez omezení. Porovnejme to s licenci většiny komerčních softwarových balíků, kde máte možnost instalovat software na jednom počítači, není dovoleno dělat žádné kopie a nikdy nevidíte zdrojový kód. Free software umožňuje neuvěřitelnou svobodu konečnému uživateli, kromě toho, protože zdrojový kód je všeobecně k dispozici, je zde větší šance odhalit chyby a opravit je.

Když je program licencován jako GNU General Public License (the GPL):

- máte právo jej používat za jakýmkoliv účelem;
- máte právo jej modifikovat a máte přístup ke zdrojovým kódům;
- máte právo program kopírovat a šířit;
- máte právo program vylepšit a uvolnit svou vlastní verzi.

Na oplátku za tato práva, máte také nějaké povinnosti. Pokud šíříte GPL program, jsou vaše povinnosti následující:

- Musíte poskytnout kopii GPL s programem, takže příjemce si je vědom svých práv v rámci licence.
- Musíte přiložit zdrojový kód nebo jej učinit volně dostupným.
- Jestliže modifikujete zdrojový kód a šíříte modifikovanou verzi, musíte jej šířit pod licenci GPL a zveřejnit modifikovaný zdrojový kód. Nesmíte použít GPL kód jako součást proprietárního programu.)
- Nesmíte omezit vydávání licenci programu za podmínek GPL. (Nemůžete vytvořit proprietární program z GPL programu.)

Více informací o GPL naleznete [zde](#).

Získání podpory: komunita Blenderu

Je volně dostupná od začátku, i když byl ještě kód uzavřený, a hodně pomohla při přijetí Blenderu. Velká, stabilní a aktivní uživatelská komunita se kolem Blenderu shromažďuje od roku 1998.

Komunita se ukázala jako klíčová koncem léta roku 2002, když se stal z Blenderu Open Source a přešel pod licenci GNU GPL.

Komunita je nyní rozdělena mezi dva široce se překrývající weby:



Vývojová komunita soustředěná kolem [webu Blender Foundation](http://www.blender.org). Zde najdete domácí stránky vývojových projektů, funkční a dokumentační fóra, repozitář CVS se zdrojovými kódy Blenderu, všechny zdroje dokumentace a tematikou podobná veřejná diskusní fóra. Vývojáři, autoři skriptů Pythonu, pisatelé dokumentace a vůbec zde můžete najít obecně kohokoli, kdo pomáhá při vývoji Blenderu.

Jít na <http://www.blender.org>



Uživatelská komunita soustředěná kolem nezávislých stránek [BlenderArtists](http://BlenderArtists.org). Zde tvůrci v Blenderu, programátoři her, nadšenci pro Blender mají své výtvoř, získávají k nim odezvy a získávají lepší pohled na funkcionalitu Blenderu. Stejně tak je možné zde najít úvody k Blenderu a znalostní bázi.

Jít na <http://www.BlenderArtists.org>

These two websites are not the only Blender resources. The Worldwide community has created a lot of independent sites, in local languages or devoted to specialized topics. A constantly updated listing of Blender resources can be found at the above mentioned sites.

## IRC chat channels

For immediate online feedback there are three IRC chat channels permanently open on [irc.freenode.net](http://irc.freenode.net). You can join these with your favorite IRC client :

- [#blenderchat](#) for general discussion of blender;
- [#blenderqa](#) for asking questions on Blender usage;
- [#gameblender](#) for discussion on issues related to game creation with Blenders included game engine.

For developers there is also :

- [#blendercoders](#) for developers to ask questions and discuss development issues, as well as a meeting each Sunday at 4 pm Netherlands time;
- [#blenderpython](#) for discussion of the python API and script development;
- [#blenderwiki](#) for questions related to editing the wiki

## Who uses Blender?

New releases of Blender are downloaded by more than a million people around the world just in the first 10 days of release. This figure spans all platforms (Windows, Linux, and MacOS) and does not include redistribution, which is fully authorized and unrestricted. We estimate there are in excess of two million users. This manual is written to serve the wide array of talented people that use Blender:

- Hobbyist/Student that just wants to explore the world of computer graphics (CG) and 3D animation
- 2-D artist that produces single image art/posters or enhances single images as an image post-processing lab
- 2-D artist or team that produces cartoon/caricature animations for television commercials or shorts (such as "The Magic of Amelia")
- 3-D artist that works alone or with another person to produce short CG animations, possibly featuring some live action (such as "Suburban Plight").
- 3-D team that produces an animated (100% CG) movie (such as "[Elephant's Dream](#)", "[Plumiferos](#)", "[Big Buck Bunny](#)").
- 3-D team that works together to produce live action movies that include some CG.

A wide range of age groups, from teenagers to oldsters use Blender, and the user community is fairly evenly divided between novice and professional graphic artists; those occasional users as well as commercial houses. We can divide the 2-D and 3-D teams that produce movies and animations further into individual job categories. Those that use Blender include:

- Director - Defines what each Scene should contain, and the action (animation) that needs to occur within that scene. Defines shots (camera takes) within that scene.
- Modeler - Makes a virtual reality. Specialties include Character, Prop and Landscapes/Stage modelers
- Cameraman, Director of Photography (DP): sets up the camera and its motion, shoots the live action, renders the output frames.
- Material Painter - paints the set, the actors, and anything that moves. If it doesn't move, they paint it anyway.
- Animation and Rigging - makes things hop about using armatures
- Lighting and Color Specialist - Lights the stage and sets, adjusts colors to look good in the light, adds dust and dirt to materials, scenes, and textures.
- Special Purpose talent - Fluids, Motion Capture, Cloth, dust, dirt, fire, explosions, you know, the fun stuff
- Editor - takes all the raw footage from the DP and sequences it into an enjoyable movie. Cuts out unnecessary stuff.

## O této příručce

This manual is a mediawiki implementation that is written by a world-wide collaboration of volunteer [authors](#). It is updated daily, and this is the English version. Other language versions are translated, generally, from this English source for the convenience of our world-wide audience. It is constantly out of date, thanks to the tireless work of some 50 or more volunteer developers, working from around the world on this code base. However, it is the constructive goal to provide you with the best possible professional documentation on this incredible package.

To assist you in the best and most efficient way possible, this manual is organized according to the creative process generally followed by 3D artists, with appropriate stops along the way to let you know how to navigate your way in this strange territory with a new and deceptively complex software package. If you read the manual linearly, you will follow the path most artists use in both learning Blender *and* developing fully animated productions:

1. Getting to know Blender = Intro, Navigating in 3d, scene mgt
2. Models = Modeling, Modifiers
3. Lighting
4. Shading = Materials, Textures, Painting, Worlds & Backgrounds
5. Animation = Basics, Characters, Advanced, Effects & Physical Sim
6. Rendering = Rendering, Compositing, Video Seq Edit
7. Beyond Blender = Extending Blender

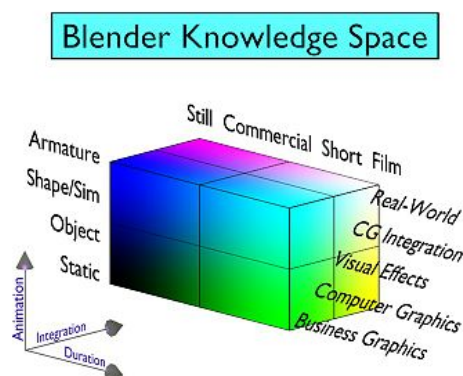
## Audience

This manual is written for a very broad audience, to answer the question "I want to *do something*; how do I do it using Blender?" all the way to "what is the latest change in the way to sculpt a mesh?"

This manual is a worldwide collaborative effort using time donated to the cause. While there may be some lag between key features being implemented and their documentation, we do strive to keep it as up-to-date as possible. We try to keep it narrowly focused on what you, the end user, need to know, and not digress too far off topic, as in discussing the meaning of life.

There are [other Blender wiki books](#) that delve deeper into other topics and present Blender from different viewpoints, such as the Tutorials, the Reference Manual, the software itself, and its scripting language. So, if a question is not answered for you in this User Manual, please search the other [Blender wiki books](#).

## Learning CG and Blender



Getting to know Blender and learning Computer Graphics (CG) are two different topics. Learning what a computer model is, and then learning how to develop one in Blender are two different things to learn. Learning good lighting techniques, and then learning about the different kinds of lamps in Blender are two different topics. The first, or conceptual understanding, is learned by taking secondary and college courses in art and media, by reading books available from the library or bookstore on art and computer graphics, and by trial and error. Even though a book or article may use a different package (like Max or Maya) as its tool, it may still be valuable because it conveys the concept.

Once you have the conceptual knowledge, you can easily learn Blender (or any other CG package). Learning both at the same time is difficult, since you are dealing with two issues. The reason for writing this is to make you aware of this dilemma, and how this manual attempts to address both topics in one wiki book. The conceptual knowledge is usually addressed in a short paragraph or two at the beginning of a topic or chapter, that explains the topic and provides a workflow, or process, for accomplishing the task. The rest of the manual section addresses the specific capabilities and features of Blender. The user manual cannot give you the full conceptual knowledge - that comes from reading books, magazines, tutorials and sometimes a lifetime of effort. You can use Blender to produce a full-length feature film, but reading this manual and using Blender won't make you another Steven Spielberg!

At a very high level, using Blender can be thought of as knowing how to accomplish imagery within three dimensions of activity:

1. Integration - rendering computer graphics, working with real-world video, or mixing the two (CGI and VFX)
2. Animation - posing and making things change shape, either manually or using simulation
3. Duration - producing a still image, a short video, a minute-long commercial, a ten minute indie short, or a full-length feature film.

Skills, like navigating in 3D space, modeling, lighting, shading, compositing, and so forth are needed to be productive in any given area within the space. Proficiency in a skill makes you productive. Tools within Blender have applicability within the space as well. For example, the video sequence editor (VSE) has very little to do with the skill of animation, but is deeply applicable along the Duration and Integration scales. From a skills-learning integration perspective, it is interesting to note that the animation curve, called an lpo curve, is used in the VSE to animate effects strips.

At the corners/intersections is where most people's interest's lie at any given time; a sort of destination, if you will. For example, there are many talented artists that produce Static-Still-CG images. Tony Mullen's book *Introducing Character Animation With Blender* addresses using CG models deformed by Armatures and shapes to produce a one-minute animation. Using Blender fluids in a TV production/commercial is at the Shape/Sim-Integrated-Minute intersection. Elephants Dream and Big Buck Bunny is a bubble at the Armature-CG-Indie space. Therefore, depending on what you want to do, various tools and topics within Blender will be of more or less interest to you.

A fourth dimension is Game Design, because it takes all of this knowledge and wraps Gaming around it as well. A game not only has a one-minute cinematic in it, but it also has actual game play, story line programming, etc. -- which may explain why it is so hard to make a game; you have to understand all this stuff before you actually can construct a game. Therefore, this Manual does not address using the Game Engine; that is a whole 'nother wiki book.

Neoficiální manuál Blender

## summary

Thanks much for blender doc activist! On the link below (or menu above) you can see whole huge "PDF reprint" of manual pages. It is usefull for travelling users, it have 1500 pages. Helpfulness is described in Czech language. Than we can see that final translation to the Czech can be very nice (if someone help).

## Co nového (october 2013)

Díky aktivitě některých přispěvovatelů v publikační činnosti wikipedie Blender vznikla možnost získat kompletní manuál Blender jako jediný soubor typu PDF.

Jde o reprint původních stránek zalámaný tak, aby byl co nejvíce čitelný a aby obsahoval kompletní "vědma" z manuálových stránek.

Výhoda tohoto manuálu spočívá v několika skutečnostech:

- vejde se na flešku, mobil, tablet. Můžete jej nosit stále u sebe
- dá se v souboru rozumně vyhledávat
- ctí strukturu manuálových stránek (jde o reprint), takže vyznáte-li se ve wiki, vyznáte se i v PDF
- je aktualizován každý měsíc(!!!) podle aktuálních anglických stránek.
- obsahuje 1500 stran textu, což je zřejmě dostatečný objem informací popisujících systém Blender :-)

Odkaz k manuálu je buď přepnutím do anglické verze horní lištou, nebo přímo na [tétu anglické originální verzi wiki](#).

## Česká verze

Bohužel je pramálo dobrovolníků, kteří pomáhají s překlady jednotlivých kapitol Blenderu 2.6, tedy je přeložena pouze hrstka stránek (prakticky obsah a sekce modelování). Nicméně blýská se na lepší časy, překládám, co mi klávesnice stačí.

Jako Vánoční (2013) překvapení pro Vás připravuji podobnou možnost získání českého "soutisku" několika kapitol tak, aby byly k užítku na cestách, zkrátka v offline použití.

Pro tisk sice anglická verze není zcela optimální (manuály a knihy mají přeci jen sazbu na vyšší grafické úrovni), nicméně je k dispozici zdarma a zkrátka JE TU!

Ω

[Quark66](#) 19:49, 24 September 2013 (CEST)(Sign)

## Instalace binárek

**Blender 2.59** je dostupný jako spustitelná binárka i jako zdrojový kód na webu Blender Foundation (<http://www.blender.org/>). V současné době pro stažení Blenderu 2.59, vyberte "Blender 2.59" z pravého menu [domovské stránky](#) .

For the online manual hosted at the wiki, you can generally use the most recent version of Blender located at the Blender Foundation website (although all of the features from the newest release version may not be fully updated). If you are using a published version of this manual it is recommended that you use the Blender version included on the Guide CD-ROM. In the following text, whenever "download" is mentioned, those using the book should instead retrieve Blender from the CD-ROM.

## Stažení a instalace binárky

Binárky jsou poskytovány pro hlavní rodiny operačních systémů. Pro další systémy mohou existovat neoficiální distribuce, tyto však nejsou podporovány [Blender Foundation](#), you should report directly to their maintainers:

- [Windows](#)
- [Linux](#)
- [MacOSX](#)
- [FreeBSD, Irix, Solaris](#)

Binárky pro operační systémy Macintosh jsou poskytovány pro dvě různé hardwarové architektury x86 (Intel a AMD processors) a PowerPC, a můžete si vybrat mezi staticky linkovanými nebo dynamicky načítanými knihovnami.

Instalátor vytvoří soubory a pár složek na dvou místech vašeho počítače: jeden set složek je pro samotný blender a další je pro uživatelská data. Pro jejich vytvoření musíte mít práva administrátora. Jsou to složky:

- .blender - configuration information (většinou výzvy ve vašem rodném jazyce)
- blendcache\_B - dočasné uložení pro data generovaná simulacemi (softbodies, cloth, fluids)
- plugins - přidávají funkcionalitu pro textury a sequencing
- scripts - pythonové skripty rozšiřují funkcionalitu Blenderu
- tmp - složka pro dočasný výstup renderu

## Hardwarová podpora

Blender supports 64-bit hardware platforms running a 64-bit unix operating system, removing the 2GB addressable memory limit.

Blender podporuje vícejádrové procesory, jako třeba Intel Core-Duo a AMD X2 tím že nabízí tzv. *Threads*: nastavují se při renderování obrázku, aby pracovaly obě jádra paralelně.

Blender podporuje širokou paletu tabletů na všech hlavních operačních systémech, především OS X, Windows XP, a Linux OS.

Tipy jak zrychlit běh Blenderu a renderování [můžete najít zde](#).

## Vývojové platformy

This is the list of systems in use and supported by active Blender developers:

Name	OS	CPU	Graphics card
Andrea Weikert	Windows XP 32	AMD Athlon 64 X2	Nvidia Quadro FX1500
Andrea Weikert	Windows XP 32	Intel P4	ATI Radeon 9000
Antony Riakiotakis	Windows 7 64	Intel Core i5	NVidia Geforce GT 540M
Antony Riakiotakis	Ubuntu 12.10	Intel Core i5	NVidia Geforce GT 540M
Bastien Montagne	Debian Testing 64	Intel Core i7 Q720	ATI Radeon 5730 mobility
Benoit Bolsee	Windows XP 32	AMD Athlon XP	ATI Radeon 9200
Brecht van Lommel	Windows 7 64	Intel Core 2 Duo	NVidia GeForce 460 GTX
Daniel Genrich	Windows Vista 64	Intel Core 2 Duo	NVidia GeForce 8500 GT
Joshua Leung	Windows Vista 32	Intel Core2 Duo	Nvidia GeForce Go 7600
Nathan Letwory	Windows 7 Ultimate 64	AMD Turion X2 Mobile RM-74	ATI HD 4650
Nathan Letwory	Windows 7 Ultimate 64	AMD Athlon II X4 620	2x HIS ATI HD 5550 /w four monitors
Robin Allen	Windows XP 32	Intel Centrino duo	NVidia GeForce go 7600
Thomas Dinges	Windows 7 x64	Intel Core i5	Intel HD 2500
Thomas Dinges	Windows 7 x64	Intel Core i7	NVidia GeForce 540M + Intel HD 3000
Andrea Weikert	Linux 32	AMD Athlon 64 X2	Nvidia Quadro FX1500
Brecht van Lommel	Linux 64	Intel Core 2 Duo	NVidia GeForce 460 GTX
Campbell Barton	Linux 64	AMD Phenom II X6	Nvidia GeForce GTS 450
Diego Borghetti	Linux 64	Intel Core i5	Nvidia GeForce GTX 480
Diego Borghetti	Linux 64	Intel Core i7	Nvidia GeForce GTX 460M

Ken Hughes	Linux 32	Intel Core Duo	Nvidia GeForce GO 7500
Ken Hughes	Linux 64	AMD Athlon 64 X2	Nvidia GeForce 6600
Kent Mein	Linux 64	Intel Core Duo	Nvidia Quadro FX 1400
Michael Fox	Linux 32	Celeron	Nvidia GeForce 6200
Raul Fernandez Hernandez	Linux 32	Pentium D 945	ATI X1550
Robin Allen	Linux 32	Intel Centrino duo	Nvidia GeForce go 7600
Brecht van Lommel	OS X 10.6	Intel Core 2 Duo	Nvidia GeForce 9600M GT
Dustin Martin	OSX 10.5	Dual Quad Intel	Nvidia Geforce 8800 GT
Ton Roosendaal	OSX 10.7	iMac Intel Core i7	AMD Radeon HD 6970M
Ton Roosendaal	OSX 10.8	MacBook Pro i7 "Retina"	Nvidia GT 650M + Intel HD 4000
Matt Ebb	OSX 10.5	Dual Core Intel MBP	nVidia 8600M
Kent Mein	SunOS 5.8	Sun Blade 150	ATI PGX
Timothy Baldrige	SGI Irix 6.5 (mipspro)	8 x R16000	(headless)
Timothy Baldrige	SGI Irix 6.5 (mipspro)	2 x R10000	
Jeroen Bakker	Latest Ubuntu 64bit	Dell m4300 Intel Core 2 Duo 2.0Ghz	Nvidia Quadro FX360M
Sergey Sharybin	Debian Wheezy 64bit	Intel Core i7 920 2.6Ghz	Nvidia GeForce GTX 560Ti + GeForce GT 620
Sergey Sharybin	Debian Wheezy 64bit	Intel Core i5 2.4GHz	Intel Sandy Bridge + Nvidia GT 520M
Jens Verwiebe	OSX 10.6/7/8/9	Intel Xeon 6-core@ 3.33	ATI 5870/7970
Nicholas Bishop	Fedora 18 64bit	Intel Core i7 @ 2.93GHz	AMD Radeon HD 6950 (Gallium drivers, currently at OpenGL 2.1)
Nicholas Bishop	Ubuntu 12.10 64bit	Intel Core i5	ATI Mobility Radeon 5650 (Gallium drivers)
Tamito Kajiyama	Windows Vista 64bit	Intel Core2 Duo	Nvidia Quadro FX 770M
Sergej Reich	Arch Linux 64bit	Intel Core2 Quad @ 2.83GHz	Nvidia GeForce GTX 285
Sergej Reich	Arch Linux 64bit	Intel Core i3 @ 2.10GHz	Intel Sandybridge Mobile
Howard Trickey	Ubuntu 12.04 64	Intel Xeon E5-1650	Nvidia Quadro 600
Howard Trickey	Windows 7 64	Intel Core i7	Nvidia GeForce GTX 460
Howard Trickey	OSX 10.8.2	Intel Core Duo	Nvidia GeForce 9400M

## Kompilace zdrojáku

There are presently four build systems for making a binary for the different operating systems supported. [See this web page](#) for more information about compiling a custom installation binary for your machine. This [link is in wiki format](#) and provides more information as well.

## Kompilace pluginů

Plugins are dynamically loaded routines that augment functionality in either texture generation or sequencing (image manipulation). See [this thread for more information](#).

## Skriptovací jazyk Python

[Python](#) je hlavní skriptovací jazyk se speciálním rozhraním, umožňující přístup ke všem vnitřním funkcím Blenderu. Skripty jsou psány v tomto jazyce, který rozšiřuje funkcionalitu Blenderu, aniž by Blender musel být překompilován. Tyto skripty píše samotný uživatel. Požadovaná verze [Python](#) je přiložená a nainstalovaná s distribucí, nicméně si jej můžete stáhnout a nainstalovat přímo z oficiálních stránek [Python website](#). Pokud stahujete Python odděleně, věnujte pozornost číslu verze. When downloading Python separately, do pay attention to the version number, as the two greatest version numbers will need to match the version numbers that Blender was compiled on, which is displayed in the console window when Blender starts. The supported python version for Blender as of the date for this edit is Python 2.6. Most functions do not rely on [Python](#) – a notable exception is the Help menu which opens a web browser pointed to a specific location. Help text is not bundled into Blender, you must download the latest wiki or PDF user manuals, found [here](#) or at [www.blender.org](http://www.blender.org).

In general, wherever you install Python, you need to establish an operating system environment variable `PYTHONPATH` and point it to the Blender Scripts directory where python modules are installed, e.g. `C:\Program Files\Blender Foundation\Blender\scripts\bpymodules` for Windows machines. Environment variables on Windows machines are set in the advanced Systems settings, in the Control Panel.

When Blender starts on a machine with Python properly installed, you will get a message in the console window, similar to this one:

```
Compiled with Python version 2.6.6.
Checking for installed Python... got it!
```

The above messages means that you have installed Python and have the full development and execution environment, and will be able to access, execute and run all Python scripts that are bundled or available for Blender. If you see a different message, such as:

```
Could not find platform independent libraries <prefix>
Could not find platform dependent libraries <exec_prefix>
Consider setting $PYTHONHOME to <prefix>[:<exec_prefix>]
'import site' failed; use -v for traceback
Checking for installed Python... No installed Python found.
Only built-in modules are available. Some scripts may not run.
Continuing happily.
```

It just means that the full Python is not available. If you want full Python functionality, refer to the Python site for installation instructions.

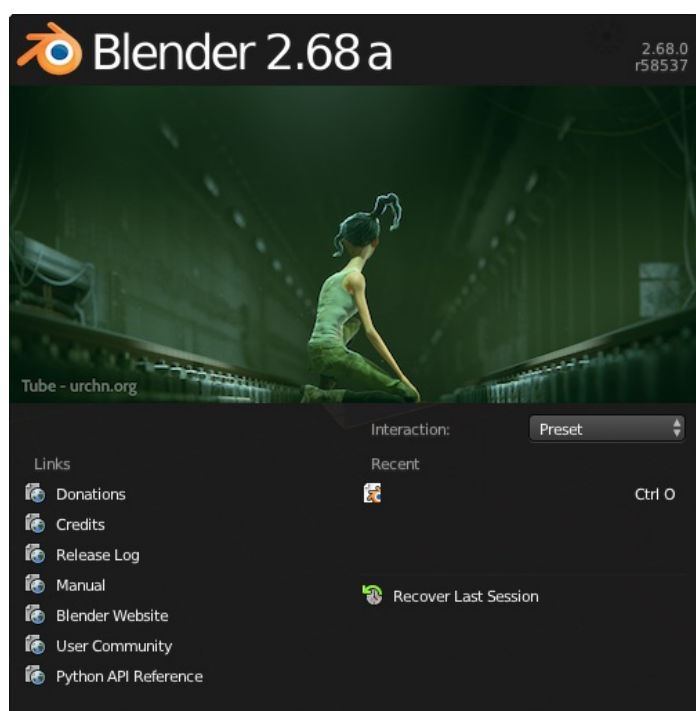
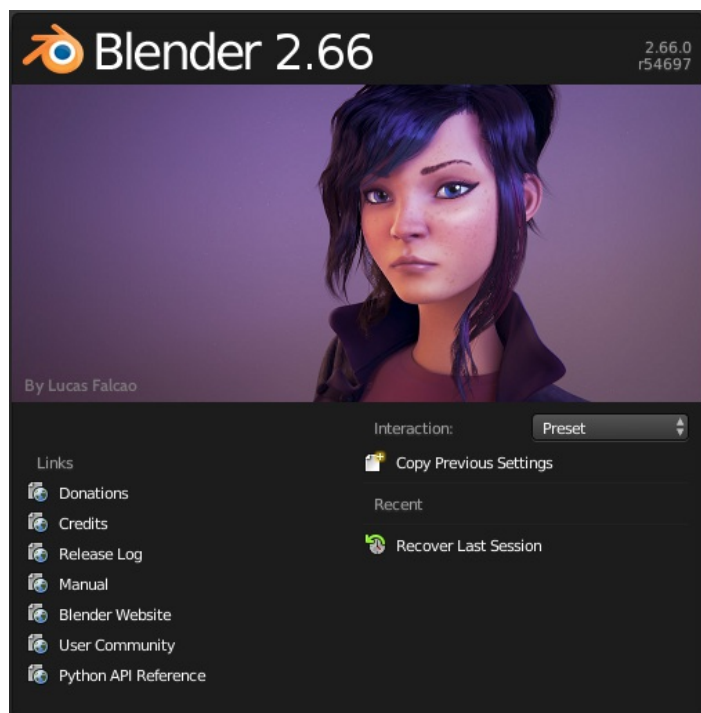
When you install Blender, you must tell the Python module where you put the scripts. If you choose to put user data in a different location for each user, then the install will put your scripts in the `C:\Users\<Current User>\AppData\Roaming\Blender Foundation\Blender\blender\scripts` folder. If you are upgrading, you probably want to overwrite all your old scripts with the new versions, and not have several versions of the same script hanging around on your PC. The best place, if you will not be editing them, is to put them in your Program Files folder with Blender:

1. Do a search on your machine for a file name with the word "scripts".
2. You will see the scripts folder appear after the initial search: `C:\Program Files\Blender-2.49\blender\scripts`, or something similar...
3. Open the script folder from the search window. You will see all the scripts. You can leave them there, or put them on your desktop temporarily...
4. Then go to Program Files, then to Blender Foundation, then blender folder, then make inside it a new folder called scripts...
5. Drag and drop or copy all the scripts from where ever you put them into this folder.
6. Make sure to include the two module folders in the script one.
7. Then, if you don't know this already, open Blender.
8. In Blender, the top menu bar hides all the preferences. Drag it down and then you will see a button marked File Paths.
9. Once you click that File Paths button, a set of path fields will be revealed.
10. Go to the Python Scripts one and type in the script folder path you just created (or use the "folder" button to open a file browser – and hit the SELECT SCRIPT PATH button to validate).
11. Then go to the File menu and Save Default Settings (CtrlU), so that Blender will remember that the script folder is where you told it to look!
12. Be careful though if you have already done stuff in Blender at this point, every time you start it it will be the default start up...

## Vaše první spuštění Blenderu

Pokud znáte Blender 2.4x, 2.5x nebo další 3D software jako je Maya, 3ds Max nebo XSI, okamžitě si všimnete, že Blender vypadá jinak, než na co jste zvyklí. Brzy si však všimnete podobností s programy, které jste už používali, jako jsou 3D Viewport, Outliner a Timeline. Pokud pracujete s 3D softwarem poprvé, můžete být trochu ztraceni. Naštěstí pro naučení 3D v Blenderu platí jediné pravidlo: nebojte se prozkoumávat a experimentovat!

Po spuštění Blenderu se podívejte na uvítací obrazovku, kde je v pravém horním rohu uvedená verze Blenderu.



Na levé straně najdete užitečné odkazy jako jsou [release log](#) verze, kterou používáte (novinky v této verzi), [wiki manuál](#) (to, co čtete právě teď) a [oficiální stránky Blenderu](#). Tyto odkazy můžete navštívit i přes menu Help.

Napravo je seznam naposledy uložených souborů (.blend). Při prvním spuštění Blenderu je seznam prázdný. Tento seznam je také dostupný přes File » Open Recent. Z rozbalovací nabídky Interaction můžete vybrat rozložení ovládání (výchozí Blender nebo Maya).

Máte tři možnosti jak začít pracovat v Blenderu:

- Klikněte na jeden z nedávných souborů (pokud jste s nějakým pracovali)
- Klikněte kamkoliv na obrazovce (mimo tmavé okno uvítací obrazovky) nebo
- Stiskněte Esc

## Průběžně ukládejte svou práci

Blender vás při ukončení programu neupozorní na neuloženou práci! Proto ukládejte pravidelně. Pokud Blender zavřete bez uložení svých posledních změn, jde něco z nich zachránit. Jednoduše otevřete znovu Blender a klikněte na Recover Last Session na uvítací obrazovce. Tento příkaz najdete i v hlavním menu File » Recover Last Session.

Dočasný soubor typu .blend

Při každém ukončení Blenderu se poslední data uloží do dočasného souboru typu .blend. Když kliknete na recover your last session, Blender nahraje data z tohoto souboru.

## Pojetí uživatelského rozhraní



Blender je vyvíjen jako multi-platformní software, což znamená, že funguje na Linuxu, Mac OS X a Windows. Protože je rozhraní Blenderu založeno na [OpenGL](#), zjistíte, že vypadá na každém operačním systému stejně.

### Tři pravidla

Uživatelské rozhraní Blenderu je založeno na třech hlavních zásadách:

- **Žádné překrývání:** Uživatelské rozhraní vám umožňuje pohled na všechny relevantní nastavení a nástroje bez nutnosti přemísťovat nebo měnit velikost oken.<sup>(1)</sup>
- **Žádné překážení:** Nástroje a možnosti rozhraní neomezují dostupnost jiných částí Blenderu. Blender nepoužívá vyskakující okna s požadavky na zadání hodnot než se něco provede.
- **Odvoditelnost:** Práce s programem by měla být maximálně neměnná a předvídatelná, co se týče práce s myší a klávesnicí.

<sup>(1)</sup>Blender 2.5 nicméně umožňuje používání více oken pro práci na více monitorech. To je výjimka pro pravidlo *Žádné překrývání*.

### Výkonné rozhraní



Rozhraní Blenderu je vykreslováno výhradně v [OpenGL](#), které umožňuje přizpůsobit rozhraní vašim potřebám. Okna a další prvky rozhraní jdou přesunout, přiblížit a jejich obsah může být přemísťován. Vaše obrazovka může být přeoorganizována přesně podle vaší chuti pro jednotlivé úkony a rozložení může být pojmenováno a uloženo.

Blender také hojně využívá klávesové zkratky pro urychlení práce. Klávesové zkratky si můžete upravit pro jejich snadnější zapamatování.

### Přehled

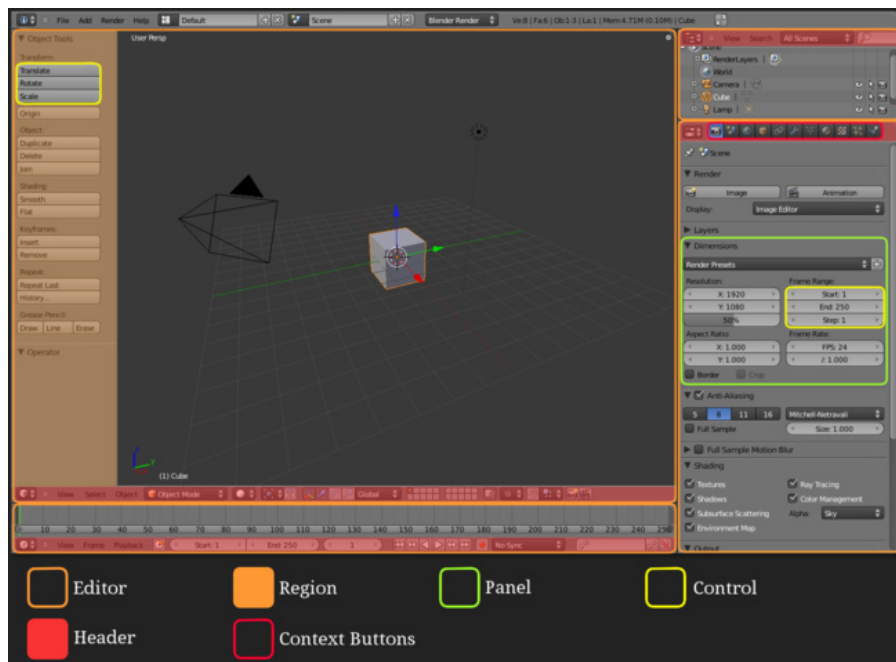
Výchozí rozhraní se skládá z Editors, Headers, Context buttons, Panels, and Controls.

- V Blenderu se **Editor** nazývá část programu, která plní specifickou funkci (3D view, Properties Editor, Video Sequence Editor, Nodes Editor...). Každý editor má nahoře nebo dole vlastní *Header*.
- **Context buttons** umožňují přístup k nastavením. Vypadají jako záložky a většinou jsou umístěny u headeru editoru (jako například u Properties Editor).
- V každém editoru jsou jednotlivá nastavení seskupena do **Panels** pro logickou organizaci rozhraní (Shadow panel, Color panel, Dimensions panel...).
- **Sidebars** jsou součástí některých editorů. Potom jsou v nich seskupeny panels a controls. Pro využití místa k práci je možné sidebars dočasně skrýt.
- Panels obsahují **Controls**. Umožňují upravit funkce, nastavení nebo hodnoty. V Blenderu je několik druhů controls :
  - **Buttons:** Umožňují přístup k nástrojům (Translate, Rotate, Insert Keyframe). Tyto nástroje mají běžně klávesovou zkratku pro urychlení práce. Pro zobrazení zkratky ukažte myší na button pro zobrazení nápovědy.
  - **Checkboxes:** Umožňují povolení nebo zakázání určité volby. Takový control může obsahovat jenom boolean hodnotu

(True/False, 1/0).

- **Sliders:** Umožňují zadání desetinných čísel ("*Floating values*"). Mohou být v určitém rozsahu (od 0.0 do 100.0) nebo neomezené (od  $-\infty$  do  $+\infty$ ). Všimněte si, že v Blenderu se vyskytují oba typy.
- **Select menus:** Umožňují výběr hodnoty ze seznamu. Od Checkboxu se liší v tom, že hodnoty jsou pojmenované a je jich víc než dvě.

[Přečtete si víc o buttons a controls »](#)



## Nastavení vstupu

Rozhraní Blenderu je navrženo k používání s níže uvedenými zařízeními a nastavením:

- Myš s kolečkem
- Klávesnice s numerickou částí
- Zapnutý NumLock




Pokud nemáte výše uvedené (např. používáte notebook), je možné tyto věci emulovat po zapnutí v nastavení Blenderu.

[Více o nastavení Blenderu »](#)



## Zápisy používané v tomto manuálu

Tento manuál používá následující zápisy popisující vstup od uživatele:

- Označení tlačítek myši:

LMB  - levé tlačítko myši  
MMB  - prostřední tlačítko myši a  
RMB  - pravé tlačítko myši

- Pokud máte myš s kolečkem

MMB  - znamená stisknutí kolečka, jako by to bylo tlačítko  
Wheel  - znamená otočení kolečkem

- Klávesové zkratky jsou zapsány obrázky kláves; například,

G - znamená malé "g".

⇧ Shift, Ctrl a Alt všeobecně mění chování jiných kláves

CtrlW or ⇧ ShiftAltA - znamená, že tyto klávesy mají být stisknuty zároveň

0 NumPad to 9 NumPad, + NumPad - a další označují klávesy na numerické části klávesnice.

Ostatní klávesy jsou označeny přímo jejich jmény Esc, ⇐ Tab, F1 až F12. Směrové šipky mají vlastní ikonky ←, → atd.

## Ovládání

Because Blender makes such extensive use of both mouse and keyboard, a golden rule has evolved among Blender users: **Keep one hand on the mouse and the other on the keyboard**. The most frequently used keys are grouped so that they can be reached by the left hand in standard position (index finger on F) on the English keyboard layout. This assumes that you use the mouse with your right hand.

Protože Blender maximálně využívá myš i klávesnici, zlaté pravidlo, které mezi uživateli Blenderu vzniklo zní: **Jednu ruku na myš, druhou na klávesnici**. Nejčastěji používané klávesy jsou seskupeny tak, že pokud položíte levý ukazovák na klávesu F při anglickém rozložení klávesnice, jde dosáhnout na všechny klávesové zkratky. To za předpokladu, že myš používáte pravou rukou.

Pokud používáte klávesnici, která se výrazně liší od anglického rozložení, pro práci v Blenderu zauvažujte nad změnou na anglické nebo americké rozložení. Klávesové zkratky i rozložení si můžete přizpůsobit. Tento manuál ale používá výchozí rozložení kláves.

[Více o nastavení Blenderu »](#)

## Emulace tlačítek myši

Pokud nemáte myš, se třemi tlačítky, budete ji muset emulovat povolením této emulace v nastavení [User Preferences](#) (ve výchozím nastavení vypnuta).

Následující tabulka ukazuje používané kombinace:

### 3-button Mouse 2-button Mouse

LMB 

MMB 

RMB 


LMB 


Alt LMB 



RMB 

### Apple Mouse

LMB  (mouse button)

⇧ Opt LMB  (Option/Alt key + mouse button)

⌘ Cmd LMB  (Command/Apple key + mouse button)

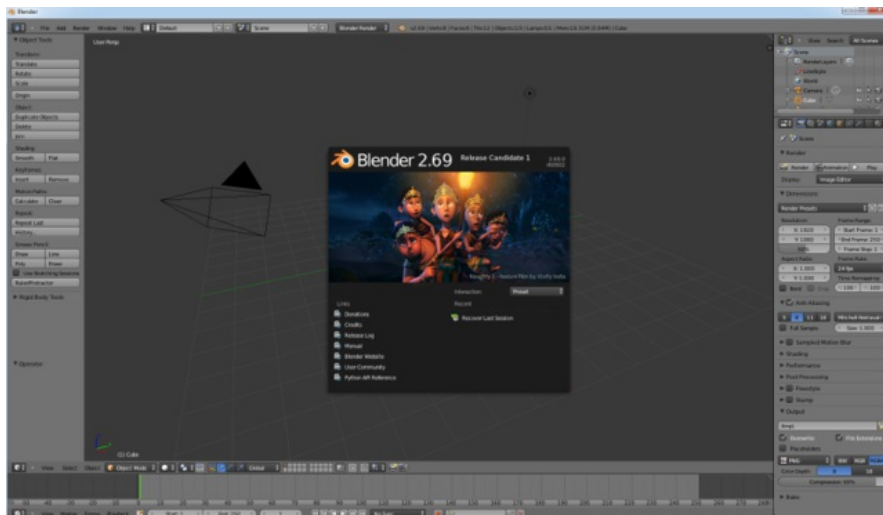
Všechny klávesové zkratky zmíněné v tomto manuálu jdou zapsat kombinacemi uvedenými v tabulce výše. Například ⇧ ShiftAlt RMB  znamená ⇧ ShiftAlt⌘ Cmd LMB  u myši s jedním tlačítkem

## Emulace NumPadu

[Více o emulaci numpadu najdete na stránce o předvolbách »](#)

## System oken

Po spuštění Blenderu by jste měli vidět následující (uvítací obrazovka uprostřed se může měnit s novými verzemi programu):

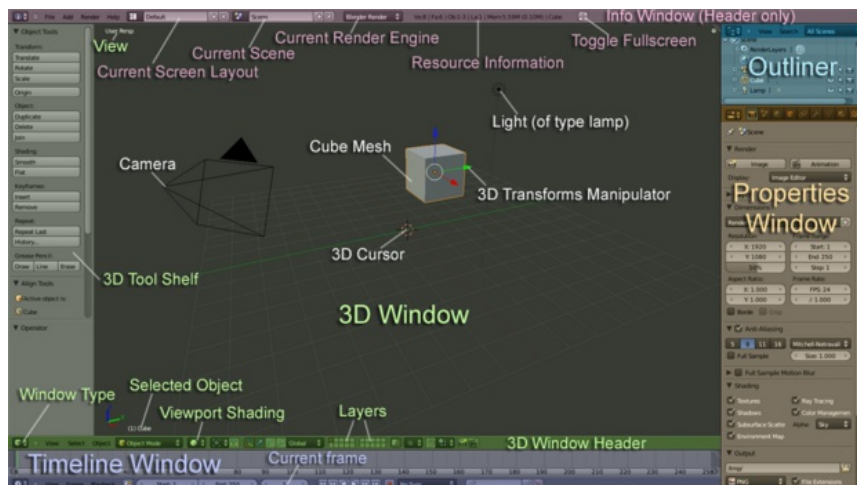


Uprostřed okna je uvítací obrazovka. Umožňuje rychlý a snadný přístup k naposledy otevřeným souborům. Pokud chcete pracovat s novým souborem jednoduše klikněte mimo uvítací obrazovku. Úvítací obrazovka zmizí a vy uvidíte výchozí rozložení a krychli.

Každé okno jde dál rozložit na samostatné části (jak je uvedeno v části [uspořádání oken](#)). Výchozí scéna je popsána níže.

## Výchozí scéna

Výchozí scéna se nahraje při každém spuštění Blenderu nebo vytvoření nového souboru.



Výchozí scéna Blenderu

Je rozdělená na pět oken:

- Hlavní menu (Info window) nahoře má jenom header (nic dalšího kromě položek menu v něm nenaleznete).
- Velké 3D okno (3D window) (3D View).
- Okno časové osy (Timeline) v dolní části programu.
- Outliner vpravo nahoře.
- Okno vlastností (Properties) (Buttons window) vpravo dole.

Na úvod popíšeme několik základních částí.

### Informační okno (Info Window) (též hlavní horní menu)


Viz. obrázek níže.

- **Current Screen (výchozí je pojmenovaná Default):** V Blenderu je ve výchozím nastavení několik přednastavených Screen ('obrazovek'), ze kterých si můžete vybrat. Pokud potřebujete [vaše vlastní uspořádání obrazovky](#), můžete si je vytvořit a pojmenovat.
- **Current Scene:** Při více [scenes \('scénách'\)](#) můžete pracovat se samostatnými virtuálními prostředními s naprosto samostatnými daty nebo s objekty a/nebo daty modelů ('mesh data') propojených mezi nimi. (V některých 3D programech obsahuje každý soubor jenom jednu scénu zatímco v Blenderu může jeden soubor obsahovat několik scén).
- **Current Rendering Engine:** Obsahuje seznam dostupných renderovacích enginů.
- **Resource Information:** Poskytují informace o Blenderu a využívaných systémových prostředcích. V této části programu se dozvíte, kolik paměti je využito v závislosti na počtu vertices, faces a objektů ve zvolené scéně, stejně jako souhrny právě vybraného. Z těchto informací můžete vyčíst, kdy se blížíte k limitům svého hardwaru.

- **Toggle Fullscreen:** Režim celé obrazovky on/off.

### 3D Window View

Viz. obrázek výše.

- **Window Type:** Umožňuje změnit typ okna (jde najít v headeru každého okna). Pokud chcete například vidět okno Outlineru můžete ho zde vybrat. Viz [Kompletní seznam typů oken](#)
- **3D Transform Manipulator:** Vizualní pomůcka pro upravování objektů (přesunutí, rotaci a změna velikosti). Objekty můžete upravovat i pomocí klávesových zkratk:: (G/R/S); CtrlSpace skryje nebo zobrazí manipulátor čehož jde dosáhnout i kliknutím na ikonu coordinate system na panelu nástrojů. Táhla pro přemístění/rotaci/změnu velikosti zobrazíte kliknutím na jejich ikony které jsou po pravé straně ikony coordinate system. • Shift LMB  při kliknutí na ikonu zobrazíte nebo skryjete jednotlivé táhla.



- **3D Cursor:** Má vícero využití. Například představuje místo, kde se vytvoří nové objekty, nebo bod, kolem kterého se bude rotovat

Tady je 3D kurzor separovaný od zbytku scény:



- **Cube Mesh** ('krychle'): Při výchozím nastavení se Blender vždy spustí s krychlí uprostřed scény (na obrázku výše byla přesunuta). Po nějaké době budete nejspíš chtít změnit "Defaultní" ('výchozí') nastavení; to můžete provést v [nastavení Blenderu](#) aby byl Blender po spuštění, tak jako ho chcete vy a uložení tohoto nastavení jako Defaultní ('výchozí') pomocí CtrlU (Save Default Settings).
- **Světlo (typu Lamp):** Při výchozím nastavení se Blender vždy spustí se zdrojem světla někde poblíž středu 3D prostoru.
- **Camera:** Při výchozím nastavení se Blender vždy spustí se Camera umístěnou někde poblíž středu 3D prostoru a namířenou směrem ke středu.
- **Selected object:** Toto pole zobrazuje jméno právě vybraného objektu.

### 3D Window Header

Toto je header 3D okna. Každé okno v Blenderu má header (a jako v tomto případě může být ve spodní části okna ale i tak mu říkáme header ('záhlaví').

Více o [Headeru v Blenderu](#) »

Viz. obrázek výše.

- **Viewport shading:** Blender vykresluje 3D okno pomocí [OpenGL](#). Druh shadingu v náhledu na scénu můžete vybrat kliknutím na tlačítko a výběrem jedné z možností. Na výběr máte od zástupných drátěných modelů až k celkovém pokrytí texturami. Pokud se chystáte používat Textured style ('náhled s texturami'), je vhodné mít výkonnou grafickou kartu.
- **Layers** ('vrstvy'): Usnadněte si modelování a animování. Vrstvy v Blenderu slouží k rozdělení objektů do funkčních skupin. Jedna vrstva například může obsahovat objekt vody, další stromy, a další zase kamery a světla. Pro zpřehlednění scény můžete vrstvy zapínat a skrývat.

### Buttons (Properties) Window Header

Header okna Properties je na obrázku vybarvený tmavěji.

Viz. obrázek výše.

Okno properties zobrazuje panely a tyto panely jsou seskupeny. V headeru tohoto okna je řada buttons ('tlačítek') (označovaných jako Context Buttons) kterými vybíráte, jaká skupina panelů se zobrazí.



Panely v okně Properties usnadňují práci seskupováním souvisejících tlačítek. To, které panely jsou viditelné závisí na typu vybraného objektu. Panely mohou být sbaleny pomocí malé šipky nalevo od názvu panelu (např. před "Render") a mohou být přeskupeny tažením za pravý horní roh.

### Outliner

Toto okno obsahuje seznam všech objektů ve scéně. Je obzvlášť užitečné při práci s většími scénami s mnoha objekty.

V headeru tohoto okna můžete zvolit typ objektů, které jsou tu zobrazeny a jak jsou v něm zobrazeny.

### Timeline Window ('Okno časové osy')

Toto okno zobrazuje časovou osu ('timeline') ve které se můžete pohybovat pomocí LMB .

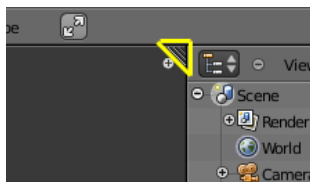
## Rozložení rámu

Blender využívá pro uspořádání ovládacích prvků v programu neotřelý systém rámu práce s nimi. Okno programu se dělí na rámy s upravitelnou velikostí. Obsah rámu je určen jeho typem jako je například 3D View nebo Outliner. Základní myšlenkou je zde možnost rozdělit jeden rám na několik menších, neustále pravoúhlých, nepřekrývajících se rámu. Díky tomu je každý rám vždy plně viditelný a je velice snadné přecházet z jednoho rámu do druhého.

## Maximalizace rámu

Pokud chcete, můžete určitý rám maximalizovat, aby vyplnil celé okno programu přes nabídku View → Toggle Full Screen daného rámu. Pro zrušení maximalizace postupujte stejně View → Toggle Full Screen. Rychlejší způsob pro přepínání mezi maximalizací určitého rámu a zobrazením všech rámu je klávesová zkratka ⇧ ShiftSpace, CtrlJ nebo Ctrl↑. POZNÁMKA: Při použití klávesové zkratky se maximalizuje ten rám, nad kterým je při stisknutí zkratky ukazatel myši.

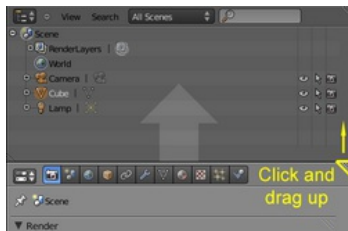
## Dělení rámu



V pravém horním rohu a v levém dolním rohu rámu se nacházejí táhla pro práci s rámy. Oboje mohou rámy dělit i slučovat. Kurzor myši se nad nimi změní na kříž. LMB a táhnete pravý horní roh doleva pro rozdělení rámu vertikálně nebo směrem dolů pro rozdělení horizontálně.

## Spojování rámu

Pokud chcete spojit dva rámy, musí mít jejich sousedící hrany stejnou délku. Pokud chcete například spojit rámy, které stojí vedle sebe, musejí mít stejnou výšku. Pokud jejich výška stejná není, nebudete je moci sloučit. Důvodem je, aby sloučené rámy měly uvnitř jenom pravé úhly. To samé platí i pro spojení rámu ležících nad sebou, musejí mít stejnou šířku. Pokud je horní rám vertikálně rozdělený na dva, musíte tyto dva horní nejprve sloučit a teprve pak půjde spojit horní rám se spodním.



Pro sloučení spodního rámu s horním (na obrázku se chystá sloučení rámu Properties s Outlinerem) umístíte kurzor nad pravé horní táhlo pro rozdělování nebo slučování rámu. Když se kurzor změní na kříž, stisknete LMB a táhnete kurzor do hořejšího rámu. Ten ztmavne a na jeho ploše se objeví šipka mířící vzhůru. Ta znamená, že spodní rám "zabere" ztmavlý prostor hořejšího rámu. Pustíte LMB a rámy se sloučí. Pokud si to rozmyslíte a chcete aby horní rám zabral prostor spodního, táhnete kurzor z horního rámu zpět do spodního a šipka ukazující směrem dolů vyplní spodní rám.

Stejným způsobem mohou být rámy slučovány zleva doprava a naopak.

Pokud stisknete Esc před uvolněním tlačítka myši, operace se zruší.

## Změna velikosti rámu

Velikosti rámu můžete měnit táhnutím za jejich okraje pomocí LMB. Umístíte kurzor nad šev, kde se stýkají dva rámy, a když se kurzor změní na dvousměrnou šipku, stisknete LMB a táhnete.

## Záměna obsahu rámu

Obsah dvou rámu můžete zaměnit umístěním kurzoru nad jedno z táhel pro slučování a dělení rámu, podržením Ctrl LMB a uvolněním tlačítka nad cílovým rámem. Tyto dva rámy nemusí být vedle sebe, ale musí být v jednom okně programu.

## Otevírání nových oken

Můžete chtít nové samostatné okno s určitým rámem. Tuto možnost využijete, pokud máte například k dispozici více monitorů a chce vidět různé informace z jedné instance Blenderu.

Vše, co pro to musíte udělat, je stisknout ⇧ Shift LMB nad táhlem pro dělení a slučování daného rámu a táhnout mimo tento rám. Objeví se nové okno i s tlačítky maximalizovat, minimalizovat, zavřít a dalšími tlačítky, podle vašeho operačního systému, obsahující jediný rám, duplikát toho, u kterého jste provedli operaci ⇧ Shift LMB a tažení.

Jakmile máte toto nové okno, můžete ho přemístit na druhý monitor (nebo si ho ponechat na tom současném); můžete u něj měnit rozměry (nebo je nechat nezměněné); můžete také organizovat jeho obsah stejnými dosud popsávanými metodami (dělit rámy a měnit jejich velikosti, přizpůsobovat si je, jak je potřeba) a tak dále.

Je i další způsob jak vyvolat nové okno. *File → User Preferences...* (nebo CtrlAltU) vyvolá nové okno s jediným rámem *User Preferences*. Pak s tímto oknem může nakládat stejně, jak bylo popsáno výše.

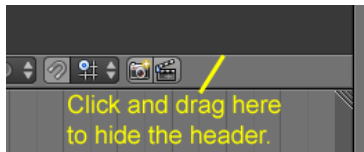
## Záhlaví oken

All windows have a header (the strip with a lighter grey background containing icon buttons). We will also refer to the header as the window *ToolBar*. The header may be at the top (as with the Properties Window) or the bottom (as with the 3D Window) of a window's area. The picture below shows the header of the 3D window:



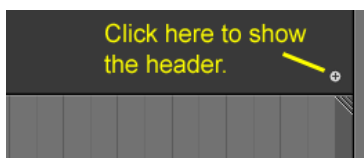
If you move the mouse over a window, its header changes to a slightly lighter shade of grey. This means that it is "focused". All hotkeys you press will now affect the contents of this window.

## Skryté záhlaví



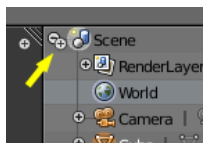
To hide a header, move your mouse over the thin line between a window and its header, until the pointer takes the form of an up-down-arrow. Then click, hold and drag with LMB from the window over the header to hide the latter.

## Odkryté záhlaví



A hidden header leaves a little plus-sign (see picture). By LMB this, the header will be reappear.

Note 1: In the 3D window there are up to two more of these little plus signs (to the top left and right of the window). Those will open panels with several tools, not a second header.



Note 2: In some windows, the mentioned plus sign can be hard to find, because it might look like a part of other icons. One example is the Outliner, in which there are other such plus signs, thus giving the one to get the header back good camouflage.

## Pozice záhlaví

To move a header from top to bottom or the other way round, simply RMB on it and select the appropriate item from the popup menu. If the header is at the top, the item text will read "Flip to Bottom", and if the header is at the bottom the item text will read "Flip to Top".



### Theme colors

Blender allows for most of its interface color settings to be changed to suit the needs of the user. If you find that the colors you see on screen do not match those mentioned in the Manual then it could be that your default theme has been altered. Creating a new theme or selecting/altering a pre-existing one can be done by selecting the [User Preferences](#) window and clicking on the Themes tab of the window.

## Tlačítko typu okna

With the LMB clicking on the first icon at the left end of a header allows selection of one of the 16 different window types. Every window frame in blender may contain any type of window. So if you want 3D views everywhere, just go ahead and change them all.

## Menu a tlačítka

Most Window Headers, located immediately next to this first "Window Type" Menu button, exhibit a set of menus which can be hidden - again with a little minus sign. So if you cannot find a menu that was mentioned somewhere, try looking for a little plus sign (once again) next to the "Window Type" button. By clicking LMB on it, the menu will come back.

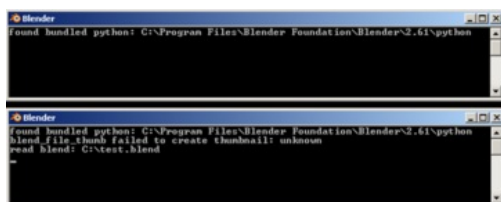
Menus allow you to directly access many features and commands, so just file through them to see what's there. All Menu entries show the relevant shortcut keys, if any.

Menus and buttons will change with Window Type and the selected object and mode. They only show the actions that can be performed.

## Okno konzole

Okno konzole je textové okno operačního systému, které zobrazuje zprávy o prováděných operacích Blenderem, jeho stav a vnitřní chyby. Pokud Blender spadne, je možné okno konzole využít jako indikátor příčiny pádu.

## Windows XP/Vista/7



Okno konzole Blender v prostředí XP spolu s hláškami.

Pokud je Blender startován v operačním systému Windows, Okno konzole se nejprve vytvoří jako samostatné okno na ploše. V případě, že jsou splněny podmínky startu, objeví se hlavní okno Blenderu a Okno konzole bude přepnuto do pozadí. Pro opětovné zobrazení přejděte do Window » Zapnout systémovou konzoli.

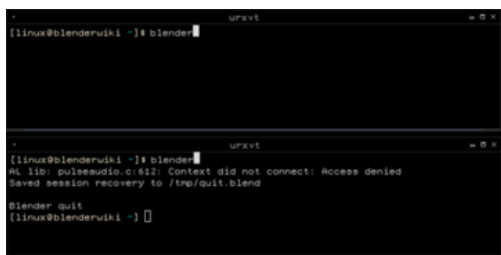
Kopie obrazovky napravo ukazuje Blenderovo okno konzole v prostředí Windows XP ihned po spuštění Blenderu a chvíli po otevření souboru spolu s příslušnými zprávami.



### Uzavření okna konzole

V případě, že Blender běží, musí okno konzole zůstat otevřené. Jeho uzavření způsobí uzavření všech oken Blenderu a tím ztráty neuložené práce. Okno konzole je velice podobné příkazovému oknu, a proto se při uzavírání oken ujistěte, že to není okno Blenderu.

## Linux



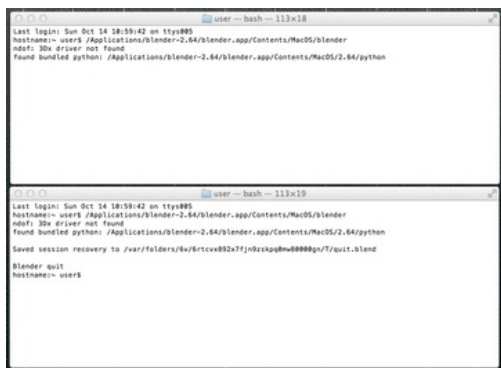
Startování Blenderu v Linuxové konzoli a následné hlášky.

Okno konzole Blenderu v Linuxu je obecně viditelné pouze v případě, je-li Blender startován z terminálového okna.

V závislosti na nastavení vašeho prostředí může být ikona Blenderu umístěna na ploše, nebo v nabídce Menu. Když spustíte Blender pomocí ikony na ploše nebo položky v nabídce menu, pravděpodobně nebude počítačové terminálové okno viditelné. [XWindows](#)

Tento brázek ukazuje Blender spuštěný z terminálu Linux /Console Window s výsledným ytištěným textem. Příklad ukazuje možnost získání přístupu k e knihovně zvukového serveru PulseAudio. Po uzavření Blenderu je obnovený soubor uložen do */tmp/quit.blend*.

## MacOS



Starting Blender from a Mac OS X console window and subsequent messages.

The process in MacOS is very similar to the one described for Linux. MacOS uses "files" with the .app extension called *applications*. These files are actually folders that appear as files in Finder. In order to run Blender you will have specify that path to the Blender executable inside this folder, to get all output printed to the terminal. You can start a terminal from Applications -> Utilities. The path to the executable in the .app folder is *./blender.app/Contents/MacOS/blender*.

If you have Blender installed in the Applications folder, the following command could be used, adapted to the particular Blender version: `/Applications/blender-2.64/blender.app/Contents/MacOS/blender`

## Stavové a chybové hlášky v okně konzole

The Blender Console Window can display many different types of Status and Error Messages. Some messages simply inform the user what Blender is doing, but have no real impact on Blender's ability to function. Other messages can indicate serious errors that will most likely prevent Blender carrying out a particular task and may even make Blender non-responsive or shut down completely. The Blender Console Window messages can also originate internally from within the Blender code or from external sources such as [Python scripts](#).

### Běžné zprávy

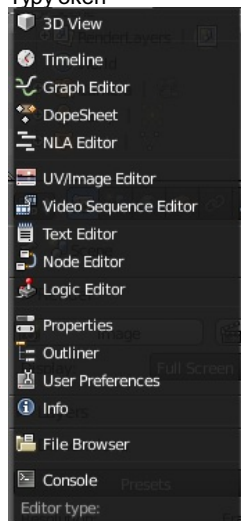
- found bundled python: (FOLDER)

This message indicates that Blender was able to find the [Python](#) library for the Python interpreter embedded within Blender. If this folder is missing or unable to be found, it is likely that an error will occur, and this message will not appear.

- malloc returns nil()

When Blender carries out operations that require extra memory (RAM), it calls a function called malloc (short for memory allocate) which tries to allocate a requested amount of memory for Blender. If this cannot be satisfied, malloc will return nil/null/0 to indicate that it failed to carry out the request. If this happens Blender will not be able to carry out the operation requested by the user. This will most likely result in Blender operating very slowly or shutting down. If you want to avoid running out of memory you can install more memory in your system, reduce the amount of detail in your Blender models, or shut down other programs and services which may be taking up memory that Blender could use.

## Typy oken



Menu pro výběr typu okna.

Grafická obrazovka Blenderu je rozdělena do několika obdélníkových částí. Každá tato část (Window Frame) může obsahovat různé typy informací podle typu okna (Window Type).

Každé okno pracuje nezávisle na ostatních a je možné mít nastavený stejný typ pro více oken současně (což bývá často například v případě 3D pohledu na scénu z různých perspektiv). Okna je možné rozdělovat a spojovat.

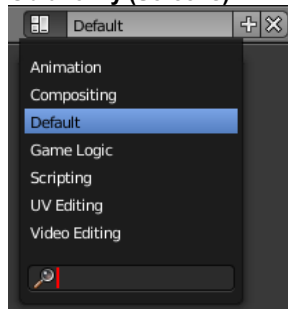
[Více o aranžování pracovních oken »](#)

Typy oken se dělí podle funkcionality:

- [The 3D View - 3D pohled](#) - otevírá grafický třírozměrný pohled na scénu.
- [The Timeline - časová osa](#) - řídí přehrávání animací.
- [The Graph Editor - grafický editor](#) - umožňuje řídit animační klíče a inter/extrapolaci chování scény kolem nich.
- [The Dope Sheet](#) - kombinuje individuální akce do akčních sekvencí.
- [The NLA Editor](#) - umožňuje řídit nelineární animační techniky sekvencí.
- [The Image/UV Editor](#) - dává nástroje k editaci mapování textur UV.
- [The Video Sequence Editor](#) - skládá videosekvence do filmového snímku.
- [The Text Editor - textový editor](#) - má možnost združovat dokumentaci k projektu, psát skripty Python apod.
- [The Node Editor - editor uzlů](#) - je využíván pro texturování materiálu a kompozice.
- [The Logic Editor](#) - Edituje logiku hry.
- [The Properties Editor - editor vlastností](#) - ukazuje různé vlastnosti právě vybraného objektu.
- [The Outliner](#) - hledá a organizuje objekty.
- [User Preferences - uživatelské nastavení](#) - je okno nastavení Blenderu jako celku.
- [Informační okno](#) - Podává informace a volby pro řízení souborů, oken a renderovacích enginů.
- [The File Browse - souborový manažer](#) - organizuje, nahrává, ukládá soubory (většinou je vyvolán automaticky podle potřeby systému).
- [The Console - konzole](#) - je přímý příkazový přístup do prostředí Python.

Je možné typ okna vybrat kliknutím na hlavičku okna *úplně* levé tlačítko. Vyskakovací menu zobrazí dostupné typy oken.

## Obrazovky (Screens)



Nabídka rozhraní

Flexibilita Blenderu vám umožňuje vytvořit a přizpůsobit si rozvržení oken tak, aby vyhovovalo různým potřebám jako je modelování, animování a skriptování. Často je užitečné se v rámci jednoho souboru rychle přepnout mezi různými prostředím. V každé scéně je třeba vymodelovat objekty, přiřadit jim materiál, otexturovat,... Na obrázku v [Systém oken](#) jsme v prostředí pro modelování.

Pro každou z těchto hlavních činností má Blender předdefinované prostředí, které jsou uzpůsobeny pro rychlou a efektivní práci. Jestliže máte potíže s nalezením konkrétní obrazovky, můžete využít vyhledávání pod seznamem. (viz obrázek vpravo).

### 1-Animation

Rozpohybování a animování herců a dalších objektů.

### 2-Model

Vytváření herců a dalších objektů.

### 3-Material

Malování a texturování povrchů.

### 4-Sequence

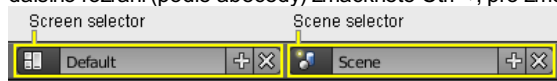
Střih a úprava scén pro film.

### 5-Scripting

Dokumentace vaší práce a skriptování.

Blender za vás automaticky setřídí Rozhraní podle abecedy a / nebo podle čísel. Přednastavená rozhraní typicky začínají číslem.

Seznam je dostupný skrz {Literal|SCR}} Menu Buttons v Info Window záhlaví zobrazeném v (*Screen and Scene selectors*). Pro změnu dalšího rozhraní (podle abecedy) zmáčkněte Ctrl→; pro změnu předchozího zmáčkněte Ctrl←.



Screen and Scene selectors

Ve výchozím nastavení si každé rozhraní 'zapamatuje' poslední scénu, která v něm byla použita. Výběrem jiného rozhraní se do něj přepneme a skočíme do scény, která v něm byla naposledy použita.

All changes to windows, as described in [Window system](#) and [Window types](#), are saved within one screen. If you change your windows in one screen, other screens won't be affected, but the scene you are working on stays the same in all screens.

## Nastavení Obrazovek

### Přidání nové obrazovky

Prostě klikněte na tlačítko "Add" (+). Na základě současného rozvržení bude vytvořeno nové rozvržení rámců.

Dejte nové obrazovce jméno *name*, které bude začínat číslem *number* tak, že jí můžete předvídatelně nalistovat použitím kurzorových kláves. Rozvržení můžete přejmenovat použitím LMB (kliknutí) v poli a napsáním nového jména, nebo opětovným kliknutím na pozici kurzoru v poli. Například můžete použít jméno "6-MyScreen". Viz: (*Screen and Scene selectors*).

### Smazání obrazovky

Obrazovku můžete smazat použitím tlačítka Delete datablock (X). Viz. (*Screen and Scene selectors*).

### Upravení Obrazovky

Použijte kontrolní prvky okna k přesunutí hranic rámců. rozdělení nebo spojení oken. Když dosáhnete rozvržení se kterým budete spokojeni, použijte CtrlU abyste uložili vaše výchozí uživatelské hodnoty (user defaults). The properties window has a special option, if you RMB (kliknutí) on its background, to arrange its panels horizontally (across) or vertically (up and down).

### Overriding Defaults

When you save a .blend file, the screen layouts are saved in it. When you open a file, the Load UI checkbox on the file browser header controls whether Blender should use the file's screen layouts, or stick with your current layouts. If Load UI is enabled, the file's screen layouts are used, overriding your defaults.

### Additional Layouts

As you get better at using Blender, consider adding some other screen layouts for a complete workflow. This will result in dramatic increases in functionality. With the dramatic increases in functionality, and as you get better at using Blender, based on what you use Blender for, consider adding some other screen layouts (for a complete workflow):

## 1-Model

4 3D windows, Properties window for Editing buttons

## 2-Lighting

3D windows for moving lights, UV/Image for displaying Render Result, Properties window for rendering and lamp properties and controls.

## 3-Material

Properties window for Material settings, 3D window for selecting objects, Outliner, Library script (if used)

## 4-UV Layout

UV/Image Editor Window, 3D Window for seaming and unwrapping mesh

## 5-Painting

UV/Image Editor for texture painting image, 3D window for painting directly on object in UV Face Select mode, 3 mini-3D windows down the side that have background reference pictures set to full strength, Properties window

## 6-Animation

Graph Editor, 3D Window for posing armature, NLA Window

## 7-Node

Big Node Editor window for noodles, UV/Image window linked to Render Result

## 8-Sequence

Graph Editor, VSE window in Image Preview mode, VSE in timeline mode, a Timeline window, and the good old Properties window.

## 9-Notes/Scripting

Outliner, Text Editor (Scripts) window

Reuse your Layouts

If you create a new window layout and would like to use it for future .blend files, simply save it as a User default by pressing CtrlU.

## Scény

Scény jsou velmi užitečným nástrojem pro správu projektů. Model krychle v prázdném prosotru, který vidíte po prvním spuštění Blenderu patří do výchozí scény. Můžete si představit, scény jsou podobné záložkám vašeho webového prohlížeče. Webový prohlížeč může mít více otevřených záložek najednou, záložky mohou být prázdné, mohou zobrazovat stejné pohledy na stejné webové stránky. Blender pracuje se scénami v podstatě stejným způsobem. Můžete mít prázdnou scénu, kompletní nezávislou kopii scény, nebo úplně novou kopii, která odkazuje na původní scénu.

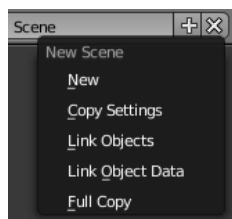
Můžete si vybrat a vytvořit scény pomocí "výběrníku" Scene selector v záhlaví okna (informační lišta v horní části většiny rozložení Blenderu, viz *obrazovka a selektory scény*).




Obrazovka a selektor scén

## Konfigurace scén

### Přidání nové scény



Menu přidání scény

Můžete přidat novou scénu kliknutím na  v možnosti detekce scén. Když vytvoříte novou scénu, můžete si vybrat z pěti možností, jak kontrolovat jeho obsah (*Add Scene menu*).

Chcete-li si vybrat mezi těmito možnostmi, je třeba jasně pochopit rozdíl mezi "objekty" a "daty objektu". Každý grafický prvek Blenderu (síťovina, lampa, křivka, atd.) se skládá ze dvou částí: objekt samotný a objekt dat (také známý jako ObData). Objekt obsahuje informace o pozici, rotaci a velikosti konkrétního prvku. ObData uchovávají informaci o síti, seznamy materiálů a tak dále. ObData je společný pro všechny instance daného typu prvku. Každý objekt má odkaz na své ObData, které jsou k němu přidružené. Naopak jeden ObData může být sdílen mnoha objekty (například jedna definice barvy a textury pro úplně odlišné a nezávislé objekty krychle, koule...)

Pět možností, které určí, "co všechno bude kopírováno" z vybrané scény do nové a jak moc bude nová scéna prolinkovaná s původní.

#### New

Vytvoří novou prázdnou scénu. V nové scéně je nastavení Renderu výchozími hodnotami.


#### Copy Settings

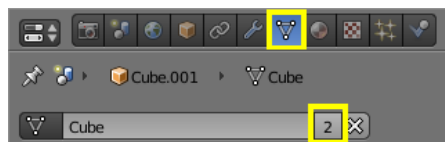
Vytvoří prázdnou scénu jako v předchozí volbě a zkopíruje nastavení vykreslování z původní do nové.

#### Link Objects

je nejmínější forma kopírování k dispozici. Tato volba vytvoří novou scénu se stejným obsahem aktuálně vybrané scény. Avšak místo kopírování objektů obsahuje nová scéna *odkazy - linky* na objekty ve staré scéně na úrovni objektů. Proto změny v *nové scéně* mají za následek stejné změny *původní scény*, protože používané objekty jsou doslova *stejně* jako v původní scéně. Vazby platí i naopak, kdy úprava objektů ve staré scéně způsobí stejné změny v nové scéně.

#### Link Object Data

Vytváří nové duplicitní kopie všech objektů v aktuálně vybrané scéně, ale každý z těchto duplicitních objektů bude mít *odkazy na ObData* (mřížky, materiály a tak podobně) odpovídající objektům v původní scéně. To znamená, že můžete změnit polohu, orientaci a velikost objektů v nové scéně bez ovlivnění jinými scénami, ale jakékoli úpravy ObData (mřížky, materiály atd.) ovlivní i další scény. To je proto, že *jeden odkaz na "ObData"* je nyní sdílen všemi objekty ve všech scénách. Chcete-li provést změny objektu v nové scéně nezávisle na objektech v jiných scénách, budete muset ručně změnit objekt v nové scéně a vytvořit "single-user" kopii pomocí LMB  a čísla v panelu Object Data (okno vlastností). Více informací najdete na stránce [typ okna](#). To má za následek vytvoření nové nezávislé kopii ObData.



#### Full Copy

Je nejhlubší forma kopírování k dispozici. Během kopie není nic sdíleno. Tato volba vytvoří úplně nezávislou scénu s kopii právě vybrané scény. Kopíruje se každý objekt spolu v odpovídající ObData.

Pro lepší pochopení, jak Blender pracuje s údaji, je vhodné pročíst [Knihovna Blenderu a datový systém](#). Také doporučuji tyto functionality "vyzkoušet" pomocí kopírování a modifikace velice jednoduchých scén.

### Stručný příklad

Uvažujme nad fimlovou scénou v baru. Můžete nejprve vytvořit bar jako čistou verzi se vším, neporušený a na svém místě. Pak se rozhodneme vytvořit akci, při níž oddělíme samostatné scény. Využití typů spojení scén může být následující:

### Link Objects

Každý objekt bude vázán k původní scéně. Pokud opravíte umístění zdi, přesune se v každé scéně která používá toto nastavení

### Link Object Data


Bude užitečné v případě, že se bude pozice objektů měnit, ale jejich tvar a nastavení zůstanou stejné. Například židle stojí na podlaze v "přeplněném baru" scény a na stolech ve scéně "Zavíráme". Vzhledem k tomu, že židle se nemění tvar, není třeba ztrácet paměť na přesných kopiích.

### Full Copy

Proces roztržení sklenice na podlaze bude potřebovat vlastní kopii, protože síť sklenice mění tvar.

Není možné dělat vše najednou ve stejné scéně, ale tyto aktivity mohou pomoci pochopit, proč je k dispozici propojení různých objektů různými způsoby.

## Odstranění scény

Můžete odstranit scénu pomocí tlačítka Delete datablock () z výběrníku (selektoru) scén (viz *Screen and Scene selectors*).

## Předvolby souborů

Obrázek nastiňuje různé převolby souborů, které jsou vysvětleny dále.



## Cesty k souborům

Pokud pracujete na důležitém projektu, je rozumné jej nakonfigurovat. Nastavení implicitních cest (paths) pro soubory různých typů.

Zde je příklad konfigurace:

<b>Fonts</b>	//fonts/
<b>Textures</b>	//textures/
<b>Texture Plugins</b>	//plugins/texture/
<b>Sequence Plugins</b>	//plugins/sequence/
<b>Render Output</b>	//renders/
<b>Scripts</b>	//scripts/
<b>Sounds</b>	//sounds/
<b>Temp</b>	//tmp/

Je vhodné poznamenat, že Blender nevytváří stromovou adresářovou strukturu automaticky. Je nutné všechny adresáře vytvořit ručně.

## Cesty ke skriptům

By default Blender looks in several directories (OS dependant) for scripts. By setting a user script path in the preferences an additional directory is looked in. This can be used to store certain scripts/templates/presets independently of the currently used Blender Version.

Inside the specified folder specific folders have to be created to tell Blender what to look for where. This folder structure has to mirror the structure of the scripts folder found in the installation directory of Blender:

```
- scripts
- addons
- modules
- presets
  - camera
  - cloth
  - interface_theme
  - operator
  - render
  - ...
- startup
- templates
```

Not all of the folders have to be present.

## Uložit & Načíst

### Relative Paths

By default, external files use a relative path. This works only when a Blender file is saved.

### Compress File

Compress .blend file when saving.

### Load UI

Default setting is to load the Window layout (the [Screens](#)) of the saved file. This can be changed individually when loading a file from the Open Blender File panel of the File Browser window.



#### File extension filter

##### Filter File Extensions

By activating this, file dialog windows will only show appropriate files (i.e. `.blend` files when loading a complete Blender setting).

The selection of file types may be changed in the file dialog window.

##### Hide Dot File/Datablocks

Hide file which start with `"."` on file browsers (in Linux and Apple systems, `"."` files are hidden).

##### Hide Recent Locations

Hides the Recent panel of the File Browser window which displays recently accessed folders.

##### Show Thumbnails

Displays a thumbnail of images and movies when using the File Browser.

## Automatické ukládání

##### Save Versions

Number of versions created for the same file (for backup).

##### Recent Files

Number of files displayed in File » Open Recent.

##### Save Preview Images

Previews of images and materials in the File Browser window are created on demand. To save these previews into your `.blend` file, enable this option (at the cost of increasing the size of your `.blend` file).

##### Auto Save Temporary File

Enable Auto Save (create a temporary file).

##### Timer

Time to wait between automatic saves.

[Read more about Auto Save options »](#)

Operace se soubory

The options to manage files are:

<a href="#">New - nový</a>	Vymaže aktuální scénu a vytvoří nový soubor pomocí startovního vzoru startup.blend
<a href="#">Open - otevřít</a>	Otevře soubor blenderu
<a href="#">Open Recent - Otevřít nedávné</a>	Nabídne k otevření seznam naposledy uložených souborů .blend
<a href="#">Recover last session - obnovit poslední sezení</a>	Natáhne soubor quit.blend který Blender ukládá automaticky při jeho ukončení. Tato volba tedy dává uživateli obnovit svůj soubor při například neočekávané chvíli pádu systému
<a href="#">Recover Auto Save</a>	This will open an automatically saved file to recover it.
<a href="#">Save - uložit</a>	Uloží aktuální soubor Blenderu
<a href="#">Save As - uložit jako</a>	Otevře průzkumník souborů a nabídne uložení souboru pod jiným jménem.
<a href="#">Save Copy - uložit kopii</a>	Uloží kopii aktuálního souboru.
<a href="#">User Preferences - uživatelské nastavení</a>	Otevře dialog uživatelských nastavení.
<a href="#">Save User Settings - uložit - uživatelské nastavení</a>	Uloží uživatelské nastavení a aktuální scénu do preferenčního souboru startup.blend.
<a href="#">Load Factory Settings - natáhnout tovární nastavení</a>	Obnoví defaultní scénu a natáhne "tovární" (poinstalační) nastavení prostředí.
<a href="#">Link or Append - připojit nebo přidat</a>	You don't have to load a complete file, you can load in only selected parts from another file if you wish.
<a href="#">Import, (COLLADA)</a>	Blender can use information stored in a variety of other format files which are created by other graphics programs.
<a href="#">Export</a>	Normally you save your work in a .blend file, but you can export some or all of your work to a format that can be processed by other graphics programs.
	Pack into .blend
	Pack all used external files into the .blend
	Unpack into Files
	Unpack all files packed into this .blend to external ones
	Make all paths Relative
	Make all paths to external files relative to current .blend
	Make all paths Absolute
	Make all paths to external files absolute
	Report Missing Files
	Report all missing external files
	Find Missing Fils
	Try to find missing external files
<a href="#">External Data</a>	

Otevírání souborů

Mód: všechny režimy

Klávesová zkratka: F1

Menu: File » Open

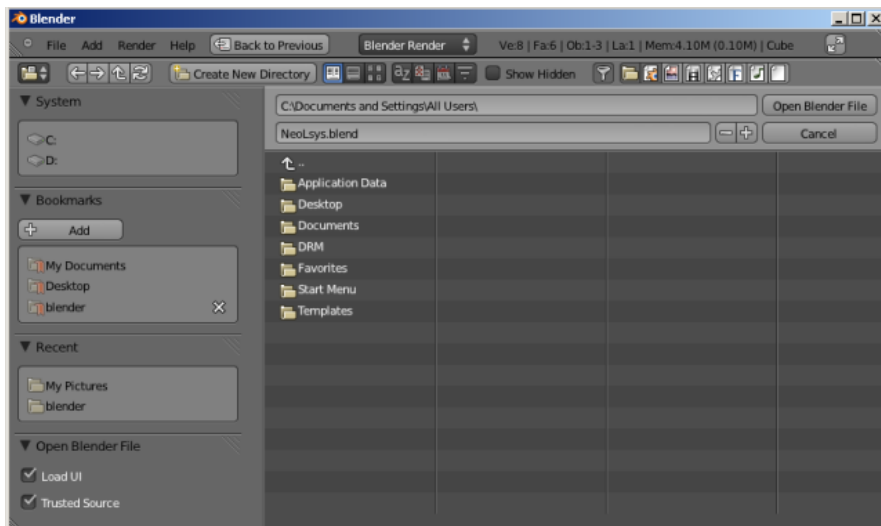
## Popis

Blender používá pro ukládání všech prvků typu objekty, scény, textury a dokonce nastavení uživatelské ho rozhraní soubory s příponou `.blend`.



Blender očekává, že vždy víte, co děláte! Pokud otevíráte soubor, **nejste** tázáni na uložení případných změn rozpracovaného právě otevřeného souboru (podobně jako při končování celé aplikace Blenderu).

**Vždy se ujistěte, zdali jste rozpracovaný soubor uložili!**



Dialogové okno otevírání souboru

## Použití správce souborů a navigace v adresářích

Pro natažení souboru Blender z disku stiskněte F1. Vyskočí okno File Browser jak je zobrazeno výše.



Horní textový blok zobrazuje cestu aktuálního adresáře a spodní text obsahuje vybraný soubor. Klávesou P (nebo tlačítkem P) se přesunete do vyššího rodičovského adresáře.

Tlačítka + a - umožňují probíhat mezi číslovanými soubory v případě, že obsahují číslo na konci svého jména.

Pro vstup do podadresáře klikněte na jeho jméno. Po kliknutí na soubor jej otevřete pomocí Open Blender File.

Kliknutím Cancel zrušíte operaci otevření souboru, uzavřete dialogové okno a vrátíte se do programu.

## Boční panel

Panel nalevo zobrazuje různé způsoby najít souboru a různé volby. Pro natažení souboru jej vyberte LMB  a po té stiskněte ↵ Enter, nebo klikněte na tlačítko Open File. Soubor může být také jednoduše natažen pomocí MMB  na jeho jméno.

## Systém

The system menu contains a list of drives that are available to navigate through to find files. Click on one to jump to that drive.

## Záložky

These are folders that you want to be able to access often without having to navigate to them in the file browser. To add a directory to the bookmark menu, navigate to that folder, then click the Add button. To remove a folder from the list, simply click the X icon next to it.

## Poslední - Recent

This is a list of recently accessed folders. You can control how many folders appear in this list by going to the File tab of the user Preferences, in the box labeled Recent Files.

## Volby otevření - Open Options

Inside each `.blend` file, Blender saves the user interface – the screen layouts. By default, this saved UI is loaded, overriding any user defaults or current screen layouts that you have. If you want to work on the blend file using your current defaults, start a fresh Blender, then open the file browser (F1). Turn off the Load UI button, and then open the file.

## Panel hlavičky

The Header contains several tools for navigation files. The four arrow icons allow you to:

- **Move to previous folder**
- **Move to next folder**
- **Move up to parent directory**
- **Refresh current folder**

Create a new folder inside the current one by clicking the Create New Directory icon.

The other icons allow you to control what files are visible and how they are displayed. You can:

- **Display files as a short list**
- **Display files as a detailed list**
- **Display files as thumbnails**

You can sort files:

- **Alphabetically**
- **By file type**
- **By Date of last edit**
- **By file size**

Filtering controls which file types are shown. Click the Enable Filtering icon, and toggle which types are shown:

- **Folders**
- **Blend files**
- **Images**
- **Movie files**
- **Scripts**
- **Font files**
- **Music files**
- **Text files**

## Další volby otevírání

From the File menu, you can also open files with the following tools:

### Open Recent

Lists recently used files. Click on one to load it in.

### Recover Last Session

This will load the `quit.blend` file Blender automatically saves just before exiting. So this option enables you to recover your last work session, e.g. if you closed Blender by accident...

### Recover Auto Save

This will open an automatically saved file to recover it.

## Bezpečnost

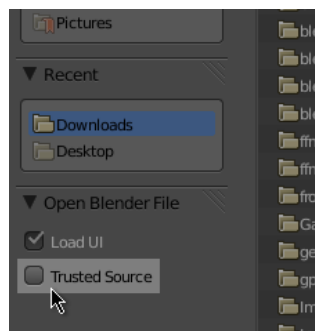
Blender is aimed at production level use and relies heavily on Python, a powerful scripting language. Python can be used in Blender to create new tools, importers and exporters, and also to drive animation rigs. With Python scripting there are endless possibilities in what you can create with Blender.

Part of Python's power comes from having full access to your system, however this power can also be misused in the wrong hands. It's possible (but not terribly likely) for dishonest people to distribute `.blend` files containing scripts that may damage your system. These scripts can be attached as part of animation rigs, so that they will be run when such a `.blend` file is opened.



***Always be very careful when downloading .blend files and tools from un-trustworthy sources!***

## Ochrana



To protect against malicious `.blend` files, it's possible to prevent any embedded scripts from running when you open a `.blend` file. This will mean that custom tools or rigs using Python features will not work, but this won't be a problem for `.blend` files that don't use these (such as material libraries), and will at least give you a chance to better evaluate what risks might be inside.

By default, Blender will trust all files and run scripts automatically. If you don't trust the file, and want protection, you can disable 'Trusted source' in the File->Open dialog in the properties section on the bottom left. Un-trusted files will disable embedded Python scripts after opening the file.

## Vrstvy (Layers)

Mód: Object Mode

Panel: Object → Draw

Klávesová zkratka: M

Menu: Object → Move to Layer...


## Popis

Scény typu 3D se často stávají nepřehledné s narůstající komplexností. Toto odstraníme tím, že jednotlivé objekty přesuneme do různých "layerů", takže se v jedné označené vrstvě zobrazí jeden objekt. Vrstvy 3D se liší od vrstev známých z 2D grafických aplikací: nemají vliv na zobrazené pořadí a jsou zde (kromě některých speciálních funkcí) výhradně pro zajištění lepší přehlednosti v modelování.

Pokud zapnete render, Blender vyrenderuje pouze ty vrstvy, které jsou označené. Pokud všechna vaše světla jsou ve vrstvě, která není 'označena', render nezobrazí nic, kromě objektů osvětlených pomocí metody ambient occlusion.

## Možnosti

### Zobrazení vrstev

Blender poskytuje 20 vrstev; můžete si vybrat, kterou chcete zobrazit pomocí malého nepopsaného tlačítka v záhlaví (*A 3D Viewport's layer buttons*). K označení pouze jedné vrstvy, klikněte na příslušné tlačítko pomocí LMB ; k označení více než jednoho tlačítka, podržte **⇧ Shift** a zaklikněte vámi požadované vrstvy.



A 3D Viewport's layer buttons.

K označení vrstev pomocí klávesnice, stiskněte 1 až 0 (na horní hlavní části klávesnice) pro vrstvy 1 až 10 (horní řada tlačítek), a Alt1 až Alt0 pro vrstvy 11 až 20 (spodní řada). Klávesa **⇧ Shift** zde též slouží pro označování více vrstev. Implicitně je v Blenderu zatrženo první horní tlačítko vrstvy.

### Přesouvání objektů mezi vrstvami



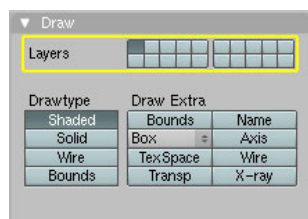
Layer selection

K přesunutí označeného objektu odlišného layeru, stiskněte M, označte požadovaný layer a z pop-up dialogu potvrďte přesun kliknutím na tlačítko OK. Objekt také může být ve více layerech najednou a toho docílíte tím, že při zobrazení pop-up dialogu podržíte **⇧ Shift** a označíte si layery, v kterých chcete, aby se objekt zobrazoval.



Object pane selection

Dalším způsobem jak zobrazit, nebo změnit zobrazení označeného objektu v layeru, je přes draw panel, který naleznete po stisknutí F7 a kliknutím na Object ikonu, viz obrázek.



Object draw pane layers

Zde, v draw panelu, jsou zobrazena tlačítka layerů, která indikují, v kterém, nebo v kterých layerech se nachází označený objekt. I zde můžete podržením **⇧ Shift** a zakliknutím požadovaných layerů určit, v kterém, nebo v kterých layerech se bude objekt zobrazovat.

### Animace vrstev

Samotné vrstvy [mohou být animovány](#). To znamená, že objekty se v jednotlivých vrstvách mohou ve scéně náhle zobrazovat a mizet.

### Příklad uspořádání vrstev

Návrh na uspořádání vrstev. Užití horní řady vrstev pro opravdu důležité věci a dolní řady pro věci, které se často nepoužívají, nemění tak často, nebo jako alternativa pro horní řadu.

1. Lead Actor
2. Supporting Actor
3. Supporting Crew (background actors)
6. Main Stage
7. Main backdrops and panels

8. Main props (tables, chairs)
9. Little props, fillers, decorations, trappings
10. Cameras, Lights
11. Lead Actor's armature
12. Supporting Actor's armature
13. Crew armatures
14. alternative clothing
15. mesh WIP
16. different stage setup, dimensions
17. different backdrops that maybe we should use
18. other big props that maybe clog up the scene
19. props WIP
20. Additional lighting

### **Skript pojmenovávající vrstvy**

Script pojmenovávající vrstvy

Je zde též dostupný skript, který umožňuje pojmenovávat jednotlivé vrstvy. Link [Layer Manager script](#)

## Modelování v Blenderu

Pokud jste shlédli kapitolu [Quick Start](#) vytváření 3D scén vyžaduje znalost alespoň tří klíčových oblastí: modely, materiály a osvětlení.

V této části se budeme hlouběji věnovat modelování. Modelování je umění a současně věda vytváření povrchů, které napodobují objekty skutečného světa, nebo vytváří imaginární a abstraktní modely.

Objekty existují ve více formách, tvarech, velikostech, a proto má Blender sadu různých nástrojů pro rychlou a efektivní práci:

### [Objekty](#)

Práce s objekty jako celky

### [Sítě](#)

Práce se sítěmi které definují tvar objektu

### [Křivky](#)

Použití křivek v modelování a ovládání objektů

### [Povrchy](#)

Modelování NURBS povrchů

### [Texty](#)

Textové nástroje pro nápisy v rozměru 3D

### [Metaobjekty](#)

Globs a Globules

### [Duplikace](#)

Duplikování, řetězení objektů

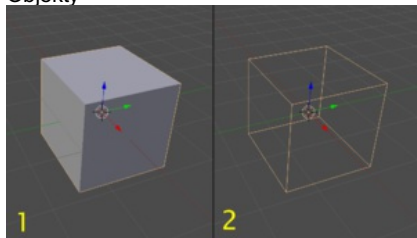
### [Modelovací skripty](#)

Přestože funkcionaliza Blenderu je rozšiřována pomocí jazyka Python, existuje zde řada velice užitečných skriptů, které mohou být při modelování užitečné.

Mnoho lidí využívá "modelování krychle" jako prvotní startovní objekt a dále pokračuje postupem protahování stěn, přesunem hran a vrcholů až k vytváření mnohem složitější sítě. Pro ploché objekty jako jsou zdi (walls) je možné využít modelování pomocí křivek ("curve modeling") které určují obrys objektu svými Bezierovskými nebo Nurbs tvary a po té je plocha roztažena do prostoru. Všechny tyto metody jsou prakticky využívány a Blender je plně svými nástroji podporuje.

*Nurbs (Non-uniform rational basis spline) je matematický model používaný pro generování a reprezentování křivek a ploch.*

## Objekty



Vybraný objekt.

Geometrie scény se skládá z jednoho nebo více objektů: například světla, křivky, povrchy, kamery, sítě a základní objekty (primitiva) popsané v části "[Primitiva sítě](#)".

## Typy objektů

[Sítě \(Mesh\)](#) Sítě jsou objekty složené z mnohoúhelníkových ploch a hran. Mohou být upravovány pomocí nástrojů Blender kategorie mesh.

[Křivky \(Curve\)](#) Křivky jsou matematicky definované objekty tak, že je s nimi možné manipulovat pomocí kontrolních bodů a linií.

[Povrchy \(Surface\)](#) Povrchy jsou obecně čtyřstranné záplaty se kterými lze manipulovat pomocí řídících bodů. Jsou užitečné pro živé, zaoblené ale jednoduché tvary..

[Metaobjekty](#) Metaball je objekt tvarovaný v prostoru 3D pomocí svého objemu. Tyto Metaball je možné využít pro vytvoření tvarů typu "Blobby" které se kvalitativně podobají tekuté látce.

[Texty](#) Objekt text je dvojrozměrným zobrazením řetězce písmen.

[Armatury.](#) Armatury jsou využívány pro kompozici třírozměrných modelů tak, aby byly jednodušejí animovatelné.

[Prázdný \(Empty\)](#) Prázdný (empty) objekt je nulový objekt určený pro vizuální transformace ostatních. Je velice užitečná například pro řízení pohybu ostatních objektů (např. pohled očí stále na tento bod).

[Kamery \(Camera\)](#) Kamera je virtuální a neviditelný objekt umožňující sledovat modelovanou scénu.

[Světla \(Lamp\)](#) Světla jsou použita pro vytvoření světelných zdrojů ve scéně.

[Force Fields](#) Force fields are used in physical simulations. They give simulations external forces, creating movement, and are represented in 3d by small control objects.



Každý objekt může být přesouván, rotován a škálován v Objektovém režimu (Object Mode). Avšak ne všechny transformace mají účinek na všechny objekty. Například škálování silového pole nezmění jeho celkový efekt



Pro další změny geometrie objektů je nutné využít editační režim (Edit mode).

Pokud jste právě přidali základní objekt, zůstáváte v objektovém režimu úprav. ~~In earlier versions of Blender, you were automatically switched into Edit mode if the Object were a Mesh, a Curve or a Surface.~~

Mezi oběma režimy úprav Object Mode a Edit Mode lze jednoduše přepínat stiskem tabulátoru ⇄ Tab.

Drátové ohraničení objektu je nyní oranžové což znamená, že je objekt vybrán a je aktivní (*Selected object*).

Obrázek výše (*Vybraný objekt*) ukazuje oba pohledy: pevný a drátový pohled na krychli. Přepnutí mezi těmito pohledy je možné pomocí klávesy Z.

## Středy objektu

Každý objekt má střed, nebo počátek. Umístění tohoto bodu předurčuje umístění objektu ve 3D prostoru,. Pokud je objekt vybrán, malé kolečko naznačuje tento střed. Umístění středu je velice důležité při rotacích a škálování objektu, viz další detaily zde [Pivot Points](#).

### Posuny středu objektu

Střed objektu může být umístěn v libovolné pozici pomocí menu objekt a pomenu transformace

- 3D Cursor Location

Přesune střed na aktuální místo 3D kurzoru. Viz [Using the 3D View](#) použití 3D kurzoru (kolečko s nitkovým křížem v prostoru).

- Střed typu Median geometrického tělesa

Přesune střed na pozici, která je průměrem jednotlivých komponent objektu.

- Geometry to Origin

Přesune střed na původní počátek objektu.

- Origin to Geometry

Přesune počátek na střed objektu.

- Origin to 3D Cursor

Přesune počátek na místo určené 3D kurzorem.

## Vymazání objektu

Mód: Edit nebo Object

Klávesová zkratka: X nebo Del

Menu: Object → Delete

Vymaže vybrané objekty.

## Spojení objektů

Mód: Object

Klávesová zkratka: CtrlJ

Menu: Object → Join Objects

Spojí všechny vybrané objekty stejného typu do jednoho objektu jehož střed je získán z předchozího *aktivního* objektu. Spojování pomocí Join je stejné jako přidávání nových objektů v editačním režimu Edit mode. Neaktivní objekty jsou smazány a zůstávají pouze aktivní objekty. Tyto operace pracují nad editovatelnými objekty typu sítě a křivky.

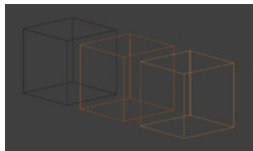
## Úvod

Výběr (selekce) jako ve většině programů určuje, se kterými prvky budeme dále pracovat. A tak je nejvýhodnější mít co nejvhodnější způsoby pro výběr prvků. V Blenderu je možností výběru vskutku hodně, a proto se jako celek nazývají *metody výběru*.

Následuje krátký popis konceptu nástrojů pro výběr, jenž jsou dostupné a dále pak několik nástrojů pro výběr objektů v objektovém režimu.

## Výběry a aktivní objekt

Blender rozlišuje mezi dvěma odlišnými stavy výběru:



Nevybraný objekt je černý, vybraný objekt oranžový a aktivní žlutý

- V objektovém režimu je poslední (ne)vybraná položka nazývána "aktivní objekt" a je zvýrazněna žlutě (ostatní oranžové). V danou chvíli je pouze jeden objekt aktivní (dokonce i když není žádný vybraný!).

Mnoho postupů v Blenderu využívá aktivní objekt jako referenční, jako například pro operace řetězení. Pokud máte vybraný objekt a potřebujete aktivní jiný objekt, jednoduše jej vyberte pomocí **⇧ RMB**.

- Všechny ostatní objekty jako tento jsou vybrány. Je možné vybrat jakýkoli počet objektů.

## Výběr bodů

Nejednoduší způsob *výběru* objektu spočívá v použití **RMB** nad ním.

Pro *přidání k výběru* použijte **⇧ RMB** nad dalšími objekty.

Pokud se *objekt překrývá* v pohledu, je možné využít **Alt RMB** a získat tak seznam všech možných voleb.

Pokud chcete *přidat k výběru* tento způsob, použijte **⇧ ShiftAlt RMB**.

Pro *aktivaci objektu* který je již vybrán klikněte **⇧ RMB** na něj.

Pro *odebrání* aktivního objektu klikněte **⇧ RMB** na něj - a dvě kliknutí, pokud objekt není aktivní. Tato funkce pracuje pouze pokud se pod myši nenachází další objekt, jinak jej přidá do výběru.

## Výběr obdélníkem

Mód: Objektový

Klávesová zkratka: B

Menu: Select → Vybrat obrys

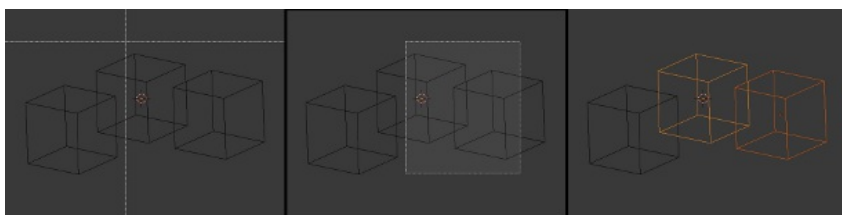
### Popis

Pomocí vybrat obrys nakreslíte obdélník držíce přitom **LMB**. Kterýkoli objekt ležící dokonce částečně uvnitř obdélníku je vybrán.

Pro odebrání objektů z výběru použijte **MMB**.

Pro zrušení vybírání použijte **RMB**.

### Příklad



Výběr obrysem ve třech krocích

In (*Vybírání obrysem...*), Vybráno je aktivováno na prvním obrázku a indikováno tečkovaným křížovým kurzorem. Na druhém obrázku je *region výběru* nakreslený pomocí **LMB**. Obdélník zahrnuje pouze dvě krychle. A nakonec na třetím obrázku výběr je kompletní uvolněním **LMB**.

Poznámka ke třetímu obrázku: světlá barva krychle nejvíce nalevo znamená "aktivní objekt", poslední vybraný objekt před použitím

nástroje výběr obrysem.

## Tipy

Výběr obrysem přidá k předchozímu výběru, a tak nejprve pomocí A odeberte z výběru vše.

## Výběr lasem

Mód: objektový

Klávesová zkratka: Ctrl LMB 

Menu: není položkano entry in the menu

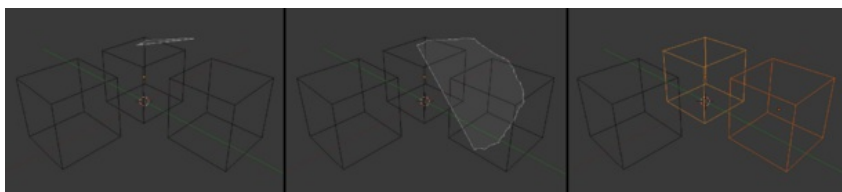
## Popis

Výběr lasem se provádí nakreslením tečkované křivky kolem hlavních bodů (pivot) objektů v režimu objektovém.

## Použití

Za současného přidržení klávesy Ctrl jednoduše nakreslíme křivku kolem středů každého objektu LMB 

Výběr typu laso přidává opět objekty již k předchozímu výběru. Pro odebrání objektů z výběru použijte Ctrl+Shift LMB 



Příklad vybírání lasem

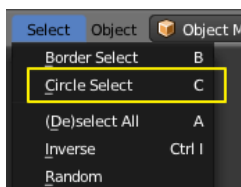
## Výběr kruhem

Mód: Objektový





Klávesová zkratka: C

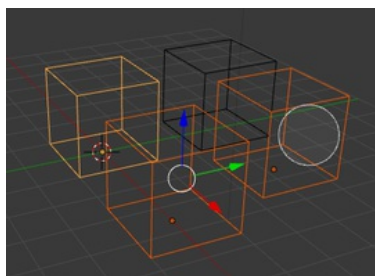
Menu: Select → Circle - výběr kruhem

## Popis

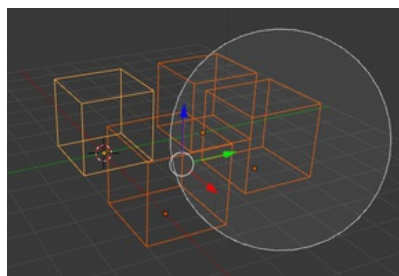


Menu pro výběr

Pomocí funkce vybrat kruhem je vykreslen kruh a pomocí držení LMB  je možné jím vybírat. Kterýkoli objekt ležící částečně uvnitř kruhu je vybrán. Velikost kruhu je možné měnit plynule pomocí MMB  jak je patrné v rozdílu mezi obrázky. Pro odebrání objektů z výběru použijte MMB . Pro zrušení vybírání použijte RMB , nebo klávesu Esc



Výběr pomocí plochy kruhu



a se zvětšeným kruhem

## Výběr pomocí menu

Výše popsané metody výběru patří do skupiny běžných. Existuje mnohem více možností výběru pomocí volby Select z menu ve 3D pohledu.

Každá je přizpůsobena k určitým operacím.

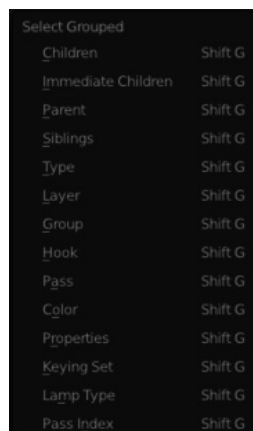
## Výběr seskupených

Mód: Objektový režim

Klávesová zkratka: ⇧ ShiftG

Menu: Select → Grouped

### Popis



Grouped menu pro výběr

Jsou dvě cesty pro organizování objektů do vzájemných vztahů. První je rodičovská vazba a druhá jednoduché seskupování (grouping). Je možné využít rodičovské vazby u nichž objekty vybíráme pomocí členů rodiny.

### Volby

Select → Grouped v Objektovém režimu používá aktivní objekt jako základ pro výběr dalších.

Dostupné možnosti jsou tyto:

#### Children

Vybere rekurzivně všechny potomky aktivního objektu.

#### Immediate Children

Vybere všechny přímé potomky aktivního objektu.

#### Parent

Vybere rodiče tohoto objektu, pokud nějakého má..

#### Siblings

Vybere objekty, které mají stejného rodiče jako aktivní objekt (sourozence). Funkce lze využít i pro výběr objektů na úrovni kořene (root).

#### Type

Vybere objekty stejného typu jako aktivní objekt..

#### Layer

Objekty, které jsou umístěny na dané vrstvě..

#### Group

Objekty, které jsou součástí skupiny v níž je aktivní objekt.

#### Object Hooks

Každý hák, který přináleží aktivnímu objektu.

#### Pass

Vybere objekty, které jsou přiřazeny ke stejnému průchodu renderem. Průchody renderem skupinami jsou nastaveny v Properties → Object → Relations a mohou být využity v Node Compositor (Add → Converter → ID Mask.)

#### Color

Vybere objekty stejné barvy Object Color. Barvy objektu jsou nastaveny pomocí Properties → Object → Display → Object Color.)

#### Properties

Vybere objekty se stejným herním engine Game Engine Properties.

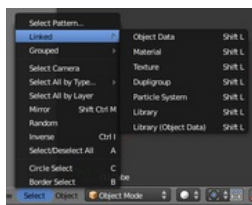
## Výběr linkovaných

Mód: Objektový režim

Klávesová zkratka: ⇧ ShiftL

Menu: Select → Linked

### Popis



Menu výběru Linked

Vybere všechny objekty které sdílí stejný datový blok jako aktivní objekt.

## Volby

Select → Linked v Objektovém režimu použije aktivní objekt jako základ pro výběr dalších.

Dostupné možnosti jsou:

### Object Data

Vybere všechny objekty, které jsou linkovány na stejná data objektu, tj. datový blok specifikující typ (síť, křivka atd.) a vytvořené objekty (konstrukční prvky jako body, řídící body v prostoru).

### Material

Vybere všechny objekty, které jsou linkovány se stejným materiálovým datablokem.

### Texture

Vybere všechny objekty, které jsou linkovány se stejným texturovým datablokem.

### Dupligroup

Vybere všechny objekty, které jsou ve stejné skupině Group pro duplikaci.

### Particle System

Vybere všechny objekty, které používají stejný částicový systém (Particle System).

### Library

Vybere všechny objekty, které jsou ve stejné knihovně ([Library](#))

### Library (Object Data)

pozn.: není popsáno v EN originále...

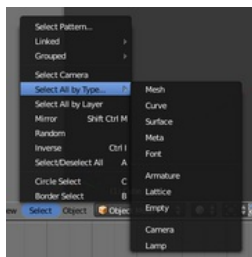
## Výběr všech podle typu

Mód: Objektový režim

Klávesová zkratka: žádná

Menu: Select → Select All by Type

## Popis



By Type selection menu

Typy jsou Mesh, Curve, Surface, Meta, Armature, Lattice, Text, Empty, Camera, Lamp.

S tímto nástrojem je možné vybrat všechny **viditelné** objekty určitého typu jednou operací.

## Volby

Select All by Type v Object režimu nabízí volbu pro objekt každého typu, který je popsán v ObData databloku.

Pouze vyberte požadované.

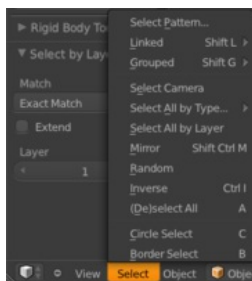
## Výběr všech podle vrstvy

Mód: Objekt režim

Klávesová zkratka: None

Menu: Select → Select All by Layer

## Popis



All by Layer menu výběru

Vrstvy jsou v jiném významu přeskupené objekty do vlastních objektů vhodné pro použití.

Tato volba umožňuje každému samotnému objektu patřícím do dané vrstvy aby byl viditelný, či nikoli pouhým jedním příkazem.

## Volby

In the Tool Shelf the following options are available:

### Extend

Enable to add objects to current selection rather than replacing the current selection.

### Layer

The layer on which the objects are.



### Výběr objektů

Lépe, nežli použití Select All by Layer může být vefektivnější vytvořit potřebné hladiny viditelné a na nich použít volbu vybrat vše A. Tato metoda umožňuje také odvybrat objekty.

## Další volby menu

Dostupné volby první úrovně menu jsou:

### Random

Náhodně vybrané na základě procentuální pravděpodobnosti v aktivní vrstvě. Výběr příkazu v numerické části je dostupný na liště Tool Shelf.

Je důležité poznamenat, že hodnota procent vyjadřuje kolik nevybraných objektů bude vybráno a nikoli kolik celkem procent objektů bude vybráno.

### Inverse (CtrlI)

Vybere všechny objekty, které nejsou vybrány zatímto odvybere objekty již vybrané.

### Select/Deselect All (A)

Pokud je nějaký objekt vybrán, odvybere jej. Jinak přepíná mezi výběrem/odvýměrem každého viditelného objektu.

### Border Select (B)

Jak je popsáno výše v části [sekce Výběr obdélníkem](#).

### Select Pattern

Vybere všechny objekty, jejichž jméno odpovídá zadané masce. Jsou podporovány zástupné znaky: \* značí cokoli, ? značí jakýkoli znak, [abc] značí jakékoli písmeno ze skupiny "abc", a ![abc] jakýkoli písmeno mimo skupinu "abc". Souhlas může být vybrán pro odpovídající velká/malá písmena, nebo ne..

Jako příklad \*domek\* odpovídá slovům se slovním základem "domek", zatímco patro\* odpovídá pouze slovům začínajícím na "patro".

## Transformace objektů

Objekty mohou být transformovány mnoha způsoby. V následujících kapitolách jsou transformace detailně popsány [Manipulace ve 3D](#). V zásadě je možné rozlišovat těchto několik skupin transformací v objektovém režimu editace:

- Posuv, translace - Translation
- Rotace - Rotation
- Měřítko, škálování - Scale
- Zrcadlení - Mirror

Skupiny a vazby

Quark66 dopisuji

Kopírování (duplikace)

Mód: editační and objektový režim

Klávesová zkratka: ⇧ ShiftD

Menu: Object → Duplicate

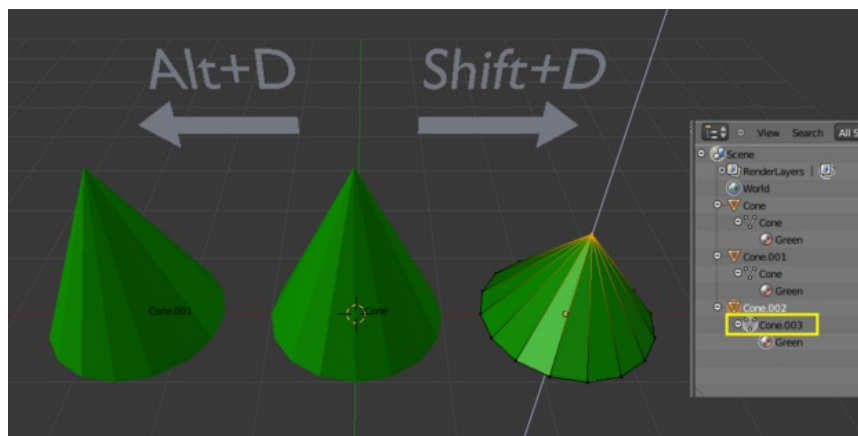
## Popis

Funkce vytvoří vizuálně identický objekt vybraného objektu(ů). Kopie je vytvořena na stejné pozici jako původní objekt a uživatel je automaticky přepnut do posuvného Grab režimu. Viž příklady níže.

Tato kopie je novým objektem, který “**sdílí**” některé datové bloky s původním objektem (materiály, textury, pohybové IPO), ale které jsou také zkopírovány jakonapříklad síťovina. Proto se také tento způsob duplikace nazývá “mělkou vazbou”, protože všechny databloky nejsou sdílené, některé jsou kopírovány stylem “hard copy”!

Note that you can choose which types of datablock will be linked or copied when duplicating: in the User Preferences' (available in the File menu) Editing “tab”, activate those types of datablocks you want to really copy in the Duplicate Data list — the others will just be linked.

## Příklady



Síť Cone.006 objektu Cone.002 je právě editovaná. Datový blok síť Unique datablock ID name je zvýrazněn v Outliner.

Kužel uprostřed je duplikován doleva (1) a také doprava (2).

- The duplicated right cone is being edited, the original cone in the middle remains unchanged. The mesh data has been copied not linked.
- Likewise, if the right cone is edited in object mode, the original cone remains unchanged. The new object's transform properties or datablock is a copy, not linked.
- When the right cone was duplicated, it inherited the material of the middle cone. The material properties were linked, not copied.

See above if you want separate copies of the datablocks normally linked.

## Linked Duplicates

Mode: Object mode

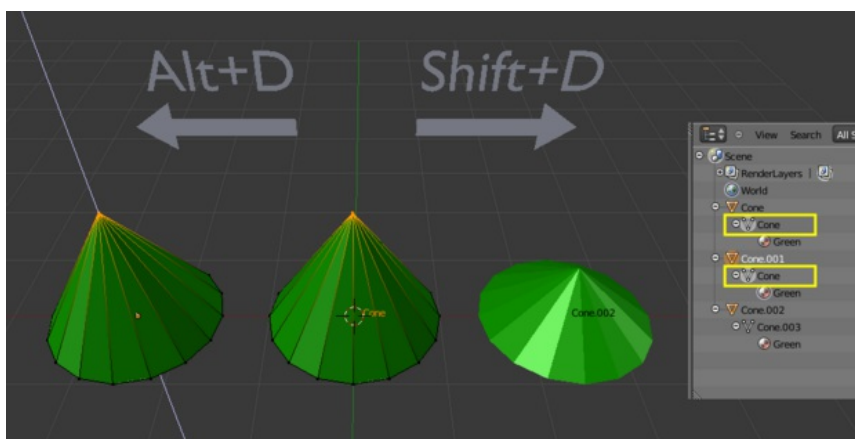
Hotkey: AltD

Menu: Object → Duplicate Linked

## Description

You also have the choice of creating a *Linked Duplicate* rather than a *Duplicate*; this is called a deep link. This will create a new object with **all** of its data linked to the original object. If you modify one of the linked objects in Edit mode, all linked copies are modified. Transform properties (object datablocks) still remain copies, not links, so you still can rotate, scale, and move freely without affecting the other copy. Reference (*Duplicate Example*) for the discussions below.

## Examples



The object Cone.001 was linked duplicated. Though both these cones are separate objects with unique names, the single mesh named Cone, highlighted in the Outliner, is shared by both.

The left cone is a Linked Duplicate of the middle cone (using AltD).

- As a vertex is moved in Edit mode in one object, the same vertex is moved in the original cone as well. The mesh data are links, not copies.
- In contrast, if one of these two cones is rotated or rescaled in object mode, the other remains unchanged. The transform properties are copied, not linked.
- As in the previous example, the newly created cone has inherited the material of the original cone. The material properties are linked, not copied.

A common table has a top and four legs. Model one leg, and then make linked duplicates three times for each of the remaining legs. If you later make a change to the mesh, all the legs will still match. Linked duplicates also apply to a set of drinking glasses, wheels on a car... anywhere there is repetition or symmetry.

## Procedural Duplication

Mode: Object mode and Edit mode

Panel: Object settings

There are currently four ways in Blender to procedurally duplicate objects. These options are located in the Object menu.

### [Verts](#)

This creates an instance of all children of this object on each vertex (for mesh objects only).

### [Faces](#)

This creates an instance of all children of this object on each face (for mesh objects only).

### [Group](#)

This creates an instance of the group with the transformation of the object. Group duplicators can be animated using actions, or can get a [Proxy](#).

### [Frames](#)

For animated objects, this creates an instance on every frame. As you'll see on this topic's subpage, this is also a very powerful technique for arranging objects and for modeling them.

## Linked Library Duplication

Hotkey: ⇧ ShiftF1

Menu: File → Link Append

### [Linked Libraries](#)

Linked Libraries are also a form of duplication. Any object or datablock in other .blend files can be reused in the current file.

## Hints

- If you want transform properties (i.e. object datablocks) to be "linked", see the page on [parenting](#).
- Material Transparency will not display when instancing dupli-groups, this is a known limitation of blenders view-port.

## Duplikace vrcholů (DupliVerts)

Mód: objektový režim

Panel: Object › Duplication

Duplikace vrcholů, nebo Duplication Vertices, či DupliVerts je duplikace základního objektu do míst vrcholů síťoviny. Jinými slovy když použijeme DupliVerts na síťovinu, vytvoří se instance základního objektu umístěné do každého vrcholu síťoviny.

Existují dva přístupy modelování pomocí duplikací DupliVerts. Mohou být využity jako aranžovací nástroj, který umožňuje sestavovat jednotlivé objekty do geometrických útvarů, (sloupy Řeského chrámu, stromy v zahradě, armáda robotů, stoly v učebně). Objekty mohou být libovolného typu, které Blender podporuje.

Druhý přístup je modelování objektu začátkem modelování jednoduché části (hroty mořského ježka, kachlíky na zdi).

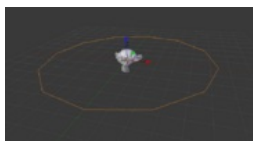
Stáhněte ukázkový soubor .blend

Můžete stáhnout příklady uvedenými na této stránce. V souboru [tento.blend](#) je první příklad" opice svázaná s kruhem na vrstvě 1, zatímco úpony připjaté je kouli na vrstvě 2.

Soubory vytvořeny v Blenderu 2.55.1 (r33567).

## Duplikace vrcholů jako aranžovací nástroj

### Nastavení



Hlava opice a kruh

Vše, co potřebujete, je základní objekt (tj. *strom*, nebo *sloup*) a nějakou síťovinu na pozadí, na jejíž vrcholy budou umístěny objekty. V této kapitole použijeme jednoduchou scénu v následující části: Použijeme hlavu opice a posadíme ji na počátek souřadného systému jako základní objekt a kruh to téhož umístění jako rodičovskou síť.



Duplikované opice

Zprvce v objektovém režimu vybereme základní objekt a pomocí **⇧ Shift RMB** přidáme kružnici k výběru (pořadí je zde velice důležité) a pomocí **Ctrl P** připojíme základní objekt ke kružnici jako k rodiči; pohneme-li kruhem, pohne se i opice.

Dále pouze s vybranou kružnicí povolíme Duplikaci vrcholů v panelu Object › Duplication › Verts. Hlava opice nyní bude umístěna do každého vrcholu kružnice.

Původní hlava opice v centru jako rodičovská síť je stále zobrazena ve 3D pohledu, ale nebude renderována. Pokud umístění a rotace hlavy není dobrá, můžeme rotace vymazat (**Alt R**), škálovat **Alt S**, přemísťovat **Alt G**, a posunout počátek (origin) **Alt O**.

### Přearanžování

Pokud nyní vyberete základní objekt a upravíte jej v editačním režimu, všechny změny budou mít vliv na tvar duplikovaných objektů. Můžete také vybrat síťovinu rodiče pro úpravu a aranžování duplikátů. Přidání vrcholů také přidá více základnímu objektu. Zde stojí za poznámku, že základní objekt bude dědit změny na rodičovské síťovině v objektovém režimu, ale nikoli v editačním režimu — tak zvětšování kružnice škálováním v objektovém režimu bude zvětšovat i velikost hlavy opice, zatím zvětšování kružnice v editačním režimu bude pouze zvětšovat vzdálenost mezi základními objekty.

### Orientace



Orientation povoleno,  
orientace +Y

Orientace základních objektů může být řízena povolením Rotace v panelu Duplication. Tím budou rotovat všechny základní objekty v souladu s normálami síťoviny rodičovského objektu.

Pro změnu orientace duplikovaných objektů vyberte základní objekt pomocí panelu Object › Relations extras a změňte směry Tracking Axes.

Výsledek různých orientací



Záporná Y

Kladná X

Kladná Z, nahoru X

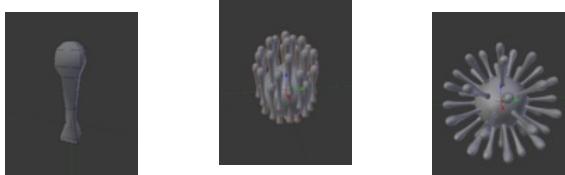
**Poznámka**

Osy objektu mohou být zviditelněny pomocí panelu Object Display.

Pro zobrazení normál sítěviny rodiče přepněte do editačního režimu a pvolte tuto funkci v panelu Properties (N) Display, kde je také možno změnit podle potřeby velikosti normál.

**Duplikace vrcholů jako modelovací nástroj**

Použitím duplikace vrcholů jako standardních primitiv je možné získat velice zajímavé modely. V tomto příkladě je vytvořeno jednoduché chapadlo pomocí několikanásobné extruze (protažení) krychle. Objekt chapadla je připojen k rodiči - icosphere. Pomocí duplikace Rotation povolíme rodičovskou sítěvinu (icosphere), orientace základního objektu (chapadlo) se přizpůsobí normálám sítěviny rodiče (v tomto případě chapadlo bylo v editačním režimu otočeno o -90° kolem osy X).



Jednoduché chapadlo  
s vyhlazením

Chapadlo duplikované  
na rodičovskou  
sítěvinu

Povolena rotace pro  
zarovnání duplikátů

Jako v předchozím případě lze tvar a proporce aranžování vylepšovat.

Pro natočení duplikátů na skutečný objekt jednoduše vyberte icosphere a použijte Object Apply Make Duplicates Real (Ctrl+ ShiftA).

Pro vytvoření jednoho objektu z icosféry a chapadla se ujistěte, že máte vše vybráno a použijte spojení Object Join (CtrlJ).

**Viz také**

Další metody duplikace jsou uvedeny [zde](#).

Duplikace stěn

Mód: Objektový režim

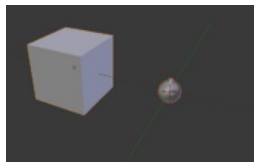
Panel: Object › Duplication

Duplikace stěn, Duplication Faces nebo DupliFaces je schopnost replikace objektů k každé stěně rodičovského objektu. Nejlepším vysvětlením pojmu je pomocí příkladů a ilustrací.



Příklad v souboru .blend

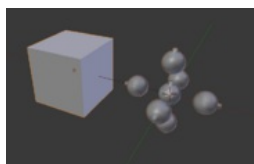
Stáhněte soubor .blend s příklady [právě zde](#)

## Základní použití



Krychle a koule

V tomto příkladu bude použita UV koule s vytaženým (extruded) "severním pólem" jako základní objekt a krychle jako rodičovská síťovina. V objektovém režimu Object mode vytvoříme kouli jako rodiče krychle. Prvně RMB  vybereme kouli, pak ⇧ Shift RMB  vybereme krychli (pořadí je velice důležité) a nakonec klávesami CtrlP provedeme jejich zpříznění.



Duplication Faces použité na krychli

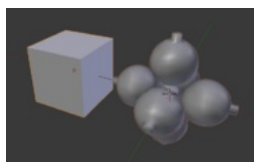
Dále v kontextu objektu a panelu Duplication povolíme povrchy Faces. Koule je zkopírována na každou plochu povrchu krychle.

Děděné vlastnosti

Umístění, orientace, škálování duplikovaných potomků odpovídá povrchovým stěnám rodiče. A tak pokud jsou různé objekty rodiči krychle, také budou duplikovány každý na stěny rodiče. Pokud je krychle rozdělena v režimu editace (Edit Mode W), každý potomek bude duplikován na každý povrch krychle.

Rodičovský objekt a originál jsou zobrazeny jako editovatelné předlohy ("templates") ve 3D pohledu, ale nejsou renderovány.

## Škálování



Scale povoleno



Horní stěna krychle škálována dolů

Povolením škálování Scale pro rodičovské objekty se také škálují potomci podle odpovídajících stěn rodičovského objektu.

Takže přeškálováním stěny rodičovského objektu budou odpovídajícím způsobem změněny i potomci.

Duplikace skupin (DupliGroup)

Mód: objektový režim

Panel: Object › Duplication › Group

Duplication Group or DupliGroup umožňuje vytvářet instance skupiny pro každou instanci objektu.

## Základní použití

- Vytvořte několik objektů a sestavte z nich skupinu pomocí
  1. výběru všech,
  2. klávesami CtrlG a
  3. eventuelně skupinu přejmenujte pomocí Object › Groups
- Vytvořte Duplikaci skupin pomocí
  1. přidání dalšího vytvořeného objektu (⇧ ShiftA), který nazveme Empty,
  2. pomocí Object › Duplication povolte skupinu Group a
  3. vyberte jméno nově vytvořené skupiny.

## Duplikace skupin a dynamické vazby

Shlédněte [Appending and Linking](#) pro pochopení dynamických vazeb (link) dat ze souboru .blend do rozpracovaného souboru. Můžete dynamicky vázat skupiny z jednoho souboru .blend do jiného. Když tak učiníte, svázaná skupina se neobjeví kdekoli ve scéně pokud nevytvoříte objekt, který skupinu bude kontrolovat.

### Příklad

- Vytvořte link skupiny nějakého souboru do vaší scény jako je popsáno v [Appending and Linking](#).  
Ot této chvíle můžete použít jednoduchý, nebo složitý způsob:
- Jenoduchá cesta:
  1. Vyberte Add › Group Instance › [jméno skupiny, kterou jste přilinkovali].
- Těžká cesta:
  1. Vyberte Add › Empty a vyberte prázdný objekt, který jste právě přidali.
  2. Přepněte na kontext Object a v panelu Duplication klikněte na Group.
  3. V rozvinutém boxu je vidět k výběru další Group: kterou jste přilinkovali.

V tomto okamžiku se projeví instance skupiny. JE možné duplikovat prázdný objekt a nastavení duplikace objektů bude zachována pro každý prázdný objekt. Touto cestou je možné velice jednoduše získat vícenásobné kopie linkovaných dat.

## Vytvoření duplikovaného objektu do reality

Pokud požadujete renderovat duplicitní skupiny v Yafaray nebo některých jiných renderech které přímo tyto duplikované skupiny nepodporují, potom:

Jednoduše vyberte duplikovanou skupinu a stiskněte Ctrl⇧ ShiftA. Po této konverzi duplicitní skupiny do regulárních objektů mohou být dále transformovány a nanimovány běžných způsobem.

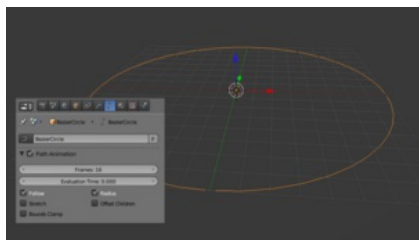
### Poznámka

Pokud duplikace skupin je linkována z externího souboru dat objektů (sítě, materiály, textury, transformace), budou také přilinkovány z původní skupiny. Avšak rozličné rodičovské vztahy se nepřenesou.

## Duplikace rámců

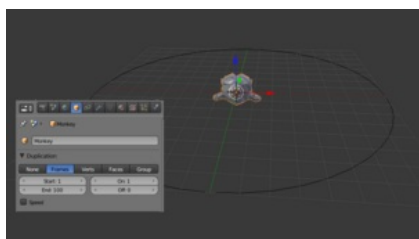
Duplikace rámců DupliFrames je nástroj pro násobné duplikace rámců rozmístěných podél dráhy. Jde o velice užitečný nástroj pro rychlé aranžování objektů

## Příklady



### Nastavení křivky

Přidání **ShiftA** Beziérovu křivku Bezier Circle a zvětšení škálováním. V menu Curve pod položkou animace cesty (Path Animation) povolte volbu sledovat enable (Follow) a nastavte hodnotu Frames na rozumnou hodnotu pod 100 (řekněme 16).

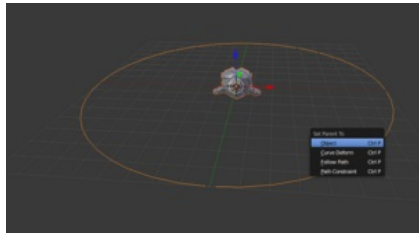


### Nastavení pro objekt

Přidání hlavy Monkey v menu Object se zapnutou možností duplikace Duplication povolte Frames a vypněte rychlost (Speed).

### Rychlost

Volba rychlosti (Speed) je využita v případě, že je nastaveno sledování cesty pro vztahy rodič-potomek pro sledování cesty Follow Path (viz níže). V tomto případě bude hlava opice cestovat podél křivky 16 rámečků.



### Rodiče

Rodičovská vazba opice na Beziérovu křivku: prvně vyberte hlavu opice, pak křivku (a tak se křivka stává aktivním objektem) a pomocí **CtrlP**. Vyberte opici a stiskněte **AltO** pro resetování jejího počátku na původní hodnotu.



### Vytažení orientace

Nyní můžete změnit orientaci opice buď jejím rotováním (buď v editačním režimu, nebo změnou v objektovém režimu) anebo změnou os posuvu Tracking Axes pod položkou Animation Hacks (s vybranou hlavou opice). Aranžování a vylepšování opic může být samozřejmě dále možné upravovat editací křivek.

Pro převod všech opic do skutečných objektů prvně **Ctrl+ShiftA** pro Make Duplicates Real. Všechny opice se stávají skutečnými objekty. Pro změnu tohoto stavu stiskněte **ObjectMake Single UserObject&Data** a pak vyberte vše pomocí **All**.

### Poznámka

Je mnoho alternativních možností přístupu k duplikaci rámců (Dupliframes). Výběr vhodného způsobu závisí na kontextu:

1. Pro použití na malých křivkách jako profilech a dlouhých jako cest jednoduše využijte tvarování Bevel Object později.
2. Pro aranžování objektů podél křivky může být užitečná kombinace modifikátoru pole (Array Modifier) a modifikátoru křivky (Curve Modifier).
3. Duplikace hran Dupliframes může být využito pro aranžování objektů jako například podél kružnice nebo pařís rozdělenou rovinou..

## Externí odkazy

- [Blender Artists: \*Dupliframes in 2.5\*](#)

Sledování (Tracking)

Page status ([reviewing guidelines](#))

## Text

vychází se zastaralé verze

Starý popis nutno doplnit v 2.6

Nutno ověřit soulad s verzí 2.6 !

==Old Track==

[http://wiki.blender.org/index.php/Doc:2.5/Manual/Modeling/Objects/Tracking#Old\\_Track](http://wiki.blender.org/index.php/Doc:2.5/Manual/Modeling/Objects/Tracking#Old_Track)

**Proposed fixes:** none

## Sledování

Mód: Objektový režim

Panel: Object » Constraints

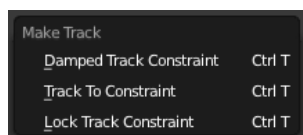
Klávesová zkratka: CtrlT

Menu: Object » Track » Make Track

## Popis

sledování se skládá s objektu sledujícího druhý objekt. Sledující "Tracker" a sledovaný cíl je "Target" (cíl). Pokud se cíl pohybuje, sledující se otáčí. Pokud se sledující pohybuje, současně se též otáčí. V obou případech sledující má stále "upřený pohled" směrem k cíli.

## Typy sledování



Make Track menu.

K vytvoření jednoho nebo více sledujících objektů jeden cíl vyberte alespoň dva objekty a stiskněte CtrlT. Aktivní objekt se stane cílem sledování ostatními objekty. Menu (*Make Track*) dává různé možnosti pro vytvoření vazby sledování:

### Sledování To Constraint

The Track To constraint applies rotations to its owner, so that it always points a given "To" axis towards its target, with another "Up" axis permanently maintained as much aligned with the global Z axis (by default) as possible. See: [Track To Constraint](#)

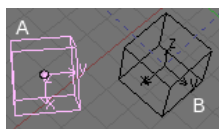
### Locked Track Constraint

The Locked Track constraint is a bit tricky to explain, both graphically and textually. Basically, it is a Track To Constraint, but with a locked axis, i.e. an axis that cannot rotate (change its orientation). Hence, the owner can only track its target by rotating around this axis, and unless the target is in the plane perpendicular to the locked axis, and crossing the owner, this owner cannot really point at its target. See: [Locked Track Constraint](#)

### Tlumené sledování

Omezení při tlumeném sledování omezuje jednu lokální osu vlastníka tak, aby pokaždé mířil směrem k cíli. Viz [Damped Track Constraint](#)

### Staré sledování

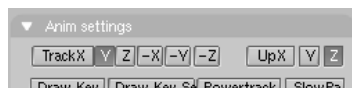


Staré sledování  
"constraint".


V tomto starším algoritmu (před verzí 2.30) je podobné chování jako Track To ve kterém není žádná osa zamčená. Tento algoritmus pouze zkouší udržet "To" osy směřující na cíl. Sledující objekt bude obvykle zkončí v podivné orientaci když je toto omezení použito poprvé. Pro opravu použijte AltR když měníte sledování "Up" osy. Avšak stále upřednostňovanou metodou je omezení typu Track To.

Předpokládejme, že nemáme vybráno nic. Old Track v dialogu se dvěma vybranými krychlemi; viz (*Old Track "constraint"*). Defaultní neaktivní objekt(y)sledují aktivní objekt a tak jejich lokální osa +Y axis ukazuje na sledovaný objekt. Kostka "A" je sledující a kostka "B" používá omezení Old Track. Uvidíte, že "A"-čková osa +Y ukazuje na "B" ale ve zvláštní orientaci. Toto se typicky stane pokud objekt již má svoji vlastní rotaci. Je možné vytvořit opravné sledování zrušením rotace sledujícího objektu, a to použitím AltR.

Orientace sledujícího objektu je také nastavena a tak je vybrána svislá osa “Up” ukazující nahoru.



Nastavení osy pro sledování.

If you want to change this you need to get to the Anim settings panel where Old Track’s settings are accessed. First select the tracking object (not the target) and change the Button window to Object context by clicking the icon () or F7; see (*Setting track axis*).

You then have the option of selecting the *Tracking axis* from the first column-set of six radio buttons and/or selecting the *upward-pointing* axis from the second column-set in the Anim Setting panel. Each time you change the “Up” axis you need to apply AltR otherwise the tracking object will continue to track with the old orientation. This is one of the drawbacks to using Old Track.

To clear or remove an old track “constraint”, select the tracking object and press AltT. As with clearing a parent constraint, you must choose whether to lose or save the rotation imposed by the tracking.

#### Note

AltT command works (and is useful) only for the Old Track “constraint”. To clear the Track To and Locked Track constraints, just delete them directly from the stack at the Constraints panel.

## Hints

The *active* object always becomes the target object to be tracked. In all but Old Track a blue dashed line is drawn between the tracker and target indicating that a tracking constraint is in place between the corresponding objects. If you see an object tracking another object without a dashed blue line then you know the tracking object is using the Old Track “constraint”.

### Invalid Tracking or settings

If you choose an invalid tracking “To” and/or “Up” axis, the tracking object keeps its current orientation and ignores the incorrect selections. For example, if you choose the +Z axis as the “To” axis and also choose the +Z axis as the “Up” axis, you have chosen an invalid combination because you can’t have the tracking object’s +Z axis doing two different things at the same time.

If you have problems setting up the correct “To” and “Up” axes you may want to turn on the tracking object’s local axes. You can do this from the Draw panel by clicking on the Axis button. See [The Interface](#) chapter for further details on the Draw panel.

## Editační mód

Editačním módem (režimem) je myšleno prostředí pracovní plochy, ve kterém přímo pracujeme s jednotlivými velice detailními částmi modelu.

## Vstup do editačního režimu

S geometrickými objekty je možné pracovat ve dvou režimech.

### Object mode

Operace v objektovém režimu se vztahují na objekt jako celek.

V Objektovém režimu má 3D pohled následující záhlaví:



Záhlaví objektového režimu

### Edit mode

Operace v editačním režimu působí na jednotlivé geometrické části objektu, ale nikoli na celkové vlastnosti jako je umístění objektu a rotace.

Editační režim má ve 3D pracovním pohledu toto záhlaví:



Záhlaví editačního režimu

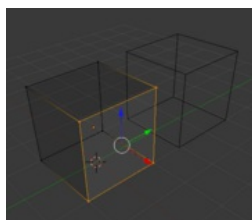
V pracovním pohledu 3D jsou k dispozici tyto nástroje a režimy (zleva doprava):

- Menu View, Select a Mesh (pohled, výběr, síťový model)
- Režim Blender
- Způsob zobrazení v pohledu 3D
- Centrum Pivot
- Manipulační 3D widget
- Režim výběru
- Přepínání hloubky bufferu
- Proporční editace
- Přichycení (Snap)
- Renderování OpenGL

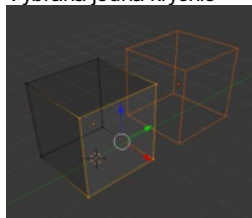
Mezi režimy je možné přepínat klávesou  $\leftrightarrow$  Tab nebo volbou požadovaného módu v záhlaví pohledu 3D.

Po vytvoření objektu je možné ihned přejít do Edit módu – a to v závislosti na přepnutí přepínače {Literal|Switch to Edit Mode}} (v uživatelském nastavení záložce User Preferences Editing). Edit mode je použitelný pouze na jeden objekt, na *aktivní*, nebo naposledy vybraný objekt.

## Vizualizace



Vybraná jedna krychle



Vybrané dvě krychle před volbou edit módu

Implicitně Blender zvýrazní vybranou geometrii oranžovou barvou pro oba režimy (Object mode a Edit mode). Tuto barvu je možné změnit uživatelským nastavením (User Preferences (CtrlAltU)→Themes.)

In Object mode with Wireframe shading enabled (Z), objects are displayed in black when unselected and in orange when selected. If more than one object is selected, all selected object except the active object, typically the object last selected, is displayed in a darker orange color. Similarly, in Edit mode, unselected geometry is drawn in black while selected faces, edges, or vertices are drawn in orange. The active face is highlighted in white.

In Edit mode, only one mesh can be edited at the time. However, several objects can be joined into a single mesh (CtrlJ in Object mode) and then separated again (P in Edit mode). If multiple objects are selected before entering Edit mode, all the selected objects remain highlighted in orange indicating that they are part of the active selection set.

If two vertices joined by an edge are selected in Vertex selection mode, the edge between them is highlighted too. Similarly, if enough vertices or edges are selected to define a face, that face is also highlighted.

## Tool Shelf



The Tool Shelf panel in edit mode (panel split in two parts for layout reasons)

Open/close the Mesh Tools panel using T. When entering Edit mode, several mesh tools become available.

Most of these tools are also available as shortcuts (displayed in the Tooltips for each tool) and/or in the Specials menu (W), the Edge menu (CtrlE), and Face menu (CtrlF). For each tool a context-dependent menu is opened at the bottom of the Tool Shelf.

Even more mesh editing tools can be enabled in the User Preferences' Add-Ons section. The development of new tools is regularly announced on Blender-related sites and forums.

For further information on panels see the [Reference panels](#) section.

## Properties Shelf



The Properties Shelf panel in edit mode (panel split in two parts for layout reasons)

Open/close the Properties Shelf using N.

In the Properties Shelf, panels directly related to mesh editing are the Transform panel, where numeric values can be entered, and the Mesh Display panel, where for example normals and numeric values for distances, angles, and areas can be turned on.

Other useful tools are found in the Properties Editor under the Object's and Object Data's Context buttons, including display options and Vertex groups.

For further information on panels see the [Reference panels](#) section.

## Zobrazení sítě (mesh)

Dodělat...

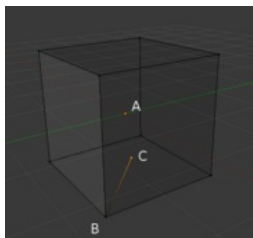
- Overlays
- Normals
- Edge/Face Info

## Vrcholy, hrany a stěny

Základní povrchové sítě objektů se skládají ze tří struktur: In basic meshes, everything is built from three basic structures: *Vrcholů*, *hran* a *stěn*. Zde nehovoříme o křivkách, zakřivených površích atd. ale není důvod se znepokojovat, zjednodušení na tři základní primitiva je naprosto dostatečné a koneckonců ve výsledku zobrazování, renderování a prostorových výpočetních operací je v určité fázi počítáno s *rozkladem* objektu na trojúhelníky s podporou tří primitiv.


*V českém překladu je využívána terminologie z oblasti analytické geometrie, tedy jednoznačně: vrchol, hrana, stěna, stejně tak jako délka, povrch (obsah [m<sup>2</sup>]), objem..*

## Vrcholy



Příklad vrcholů

Vrchol je z pohledu analytické geometrie pouhým bodem, který je možné popsat souřadnicemi v prostoru 3D. Obvykle je neviditelný (tzv. *nekonečně malý*) při procesu renderování a v objektovém režimu modelování. Nezaměňujte centrální bod objektu, nebo těžiště objektu s vrcholem. Lze jej totiž na rozdíl od zmíněných virtuálních pomyslných bodů vybrat a přímo s ním provádět například operace posuvu v prostoru. Obrázek *Příklad vrcholů* zobrazuje centrální bod tělesa označený jako "A", zatímco body "B" a "C" jsou vrcholy.

Jednoduchá cesta k vytvoření nového vrcholu je kliknutím Ctrl LMB  v Editacním módu. Samozřejmě díky dvojrozměrnosti obrazovky nemůže Blender určit všechny tři souřadnice nového bodu, a tak je nový bod umístěn do úrovně hloubky nastaveného kurzoru 3D. Použitím níže popsaných metod je možné tento bod propojit v dalším hranou. Na obrázku výše je nový vrchol označen písmenem C a nově vytvořená hrana přidaná do krychle je mezi body B a C.

## Hrany

An edge always connects two vertices by a straight line. The edges are the "wires" you see when you look at a mesh in wireframe view. They are usually invisible on the rendered image. They are used to construct faces. Create an edge by selecting two vertices and pressing F.

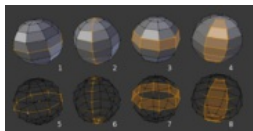
## Stěny

A face is the highest level structure in a mesh. Faces are used to build the actual surface of the object. They are what you see when you render the mesh. A face is defined as the area between either three (triangles) or four (quadrangles) vertices, with an edge on every side. Triangles are always flat and therefore easy to calculate. On the other hand, quadrangles "deform well" and are therefore preferred for subdivision modeling.

Dbejte opatrnosti při použití čtyřúhelníků v prostoru (quads), protože interně je Blenderem tento objekt dělen na dva trojúhelníky. Čtyřúhelníky mají význam v površích, které jsou ploché, lehce vyduuté.

While you could build a cube with triangular faces, it would just look more confusing in Edit mode. An area between three or four vertices, outlined by edges, doesn't have to be a face. If this area does not contain a face, it will simply be transparent or non-existent in the rendered image. To create a face, select three or four suitable vertices and press F.

## Smyčky (loops)



Smyčky hran a stěn

Edge and Face Loops are sets of faces or edges that form continuous "loops" as shown in (*Edge and Face Loops*). The top row (1-4) shows a solid view, the bottom row (5-8) a wireframe view of the same loops.

Note that loops 2 and 4 do not go around the whole model. Loops stop at so called poles because there is no unique way to continue a loop from a pole. Poles are vertices that are connected to either three, five, or more edges. Accordingly, vertices connected to exactly one, two or four edges are not poles.

In the image above, loops that do not end in poles are cyclic (1 and 3). They start and end at the same vertex and divide the model into two partitions. Loops can be a quick and powerful tool to work with specific, continuous regions of a mesh and are a prerequisite for organic character animation. For a detailed description of how to work with loops in Blender, please refer to the Manual page on [Edge and Face Tools](#).

## Smyčky hran

Loops 1 and 2 in (*Edge and Face Loops*) are edge Loops. They connect vertices so that each one on the loop has exactly two neighbors that are not on the loop and placed on both sides of the loop (except the start and end vertex in case of poles).

Edge Loops are an important concept especially in organic (subsurface) modeling and character animation. When used correctly, they allow you to build models with relatively few vertices that look very natural when used as subdivision surfaces and deform very well in animation.

Take (*Edge Loops in organic modeling*) as an example: the edge loops follow the natural contours and deformation lines of the skin and the underlying muscles and are more dense in areas that deform more when the character moves, for example at the shoulders or knees.

Further details on working with Edge Loops can be found in [Edge Loop Selection](#).

## Smyčky ploch

These are a logical extension of Edge Loops in that they consist of the faces between two Edge Loops, as shown in loops 3 and 4 in (*Edge and Face Loops*). Note that for non-circular loops (4) the faces containing the poles are not included in a Face Loop.

Further details on working with Face Loops can be found in [Face Loop Selection](#).

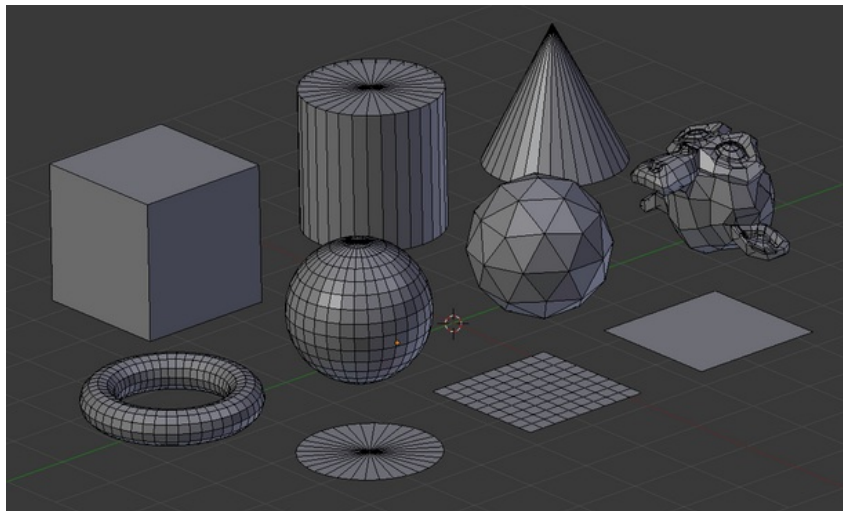
Primitiva síť

Mód: Object režim

Klávesová zkratka:  $\diamond$  ShiftA

Menu: Add » Mesh

Běžný typ objektu ve 3D scéně Blenderu je síť. Blender nabízí několik základních "primitiv", tj. tvarů, ze kterých můžeme začít modelovat.



Deset standardních primitiv Blenderu

Volby pro více než jeden objektu jsou:

Radius - poloměr

Nastaví počítáčení velikosti Kruhu - Circle, Válce - Cylinder, Kuželu - Cone, Koule - UVSphere a IcoSphere.

Hloubka - Depth

Nastaví počítáčení délku (výšku, hloubku) válce a kuželu.

Pounámka k rovinným útvarům

Můžete vytvořit z rovinného útvaru 3rozměrný posunem jedné, nebo více hran mimo základní rovinu (použitelné pro rovinu, kruh a mříž - Grid). Jednoduchá kružnice se nabízí jako výborný startovní krok pro vytváření mnoha komplexních tvarů.

## Rovina - Plane

Rovina, čili "planum" je v počátku ohraničena čtyřmi hranami a obsahuje jednu plochu. Nejde však o skutečný třírozměrný objekt, protože nemá tloušťku. Roviny jsou často využity pro tvorbu dalších objektů, nebo samostatně pro zrcadla, krycí desky s jedním povrchem atp.

## Krychle

A standard cube contains eight vertices, twelve edges, and six faces, and is a real three-dimensional object. Objects that can be created out of cubes include dice, boxes, or crates.

## Kruh

A standard circle is comprised of  $n$  vertices. The number of vertices and radius can be specified in the context panel in the Tool Shelf which appears when the circle is created.

Vertices

The number of vertices that define the circle. The more vertices the circle contains, the smoother its contour will be; see (*"Circles" obtained with various settings*). In contrast, a circle with only 3 vertices is actually a triangle — the circle is actually the standard way of adding polygons such as triangles, pentagons, et cetera.

Radius

Nastaví poloměr kružnice.

Typ výplně - Fill Type

Set how the circle will be filled

Triangle Fan

Fill with triangular faces which share a vertex in the middle.

Ngon

fill with a single ngon

Nothing

Do not fill. Creates only the outer ring of vertices

## Koule - UV Sphere

A standard UV sphere is made out of  $n$  segments and  $m$  rings. The level of detail and radius can be specified in the context panel in the Tool Shelf which appears when the UV sphere is created. Increasing the number of segments and rings makes the surface of the UV sphere smoother.

### Segments

Number of vertical segments. Like Earth's meridians, going pole to pole and

### Rings

Number of horizontal segments. These are like Earth's parallels.

### Note

If you specify a six segment, six ring UVsphere you'll get something which, in top view, is a hexagon (six segments), with five rings plus two points at the poles. Thus, one ring fewer than expected, or one more, if you count the poles as rings of radius 0.

## Koule - Icosphere

An icosphere is a polyhedra sphere made up of triangles. The number of subdivisions and radius can be specified in the context panel in the Tool Shelf after the Icosphere is created. Icospheres are normally used to achieve a more isotropical and economical layout of vertices than a UV sphere.

### Subdivisions

How many recursions are used to define the sphere. Increasing the number of subdivisions makes the surface of the Icosphere smoother. At level 1 the Icosphere is an icosahedron, a solid with 20 equilateral triangular faces. Any increasing level of subdivision splits each triangular face into four triangles, resulting in a more spherical appearance.

### Size

The radius of the sphere.

### Note

It is possible to add an icosphere subdivided 500 times. Adding such a dense mesh is a sure way to cause a program crash. An icosphere subdivided 10 times would have 5,242,880 triangles, so be very careful about this!

## Válec

A standard cylinder is made out of  $n$  vertices. The number of vertices in the circular cross-section can be specified in the context panel in the Tool Shelf that appears when the object is created; the higher the number of vertices, the smoother the circular cross-section becomes. Objects that can be created out of cylinders include handles or rods.

### Vertices

Then number of vertical edge loops used to define the cylinder.

### Radius

Sets the radius of the cylinder.

### Depth

Sets the height of the cylinder.

### Cap Fill Type

Similar to circle (see above). When set to none, the created object will be a tube. Objects that can be created out of tubes include pipes or drinking glasses (the basic difference between a cylinder and a tube is that the former has closed ends).

## Kužel

A standard cone is made out of  $n$  vertices. The number of vertices in the circular base, dimensions and option to close the base of the cone can be specified in the context panel in the Tool Shelf that appears when the object is created; the higher the number of vertices, the smoother the circular base becomes. Objects that can be created out of cones include spikes or pointed hats.

### Vertices

The number of vertical edge loops used to define the cone.

### Radius 1

Sets the radius of the base of the cone.

### Radius 2

Sets the radius of the tip of the cone. A value of 0 will produce a standard cone shape.

### Depth

Sets the height of the cylinder.

### Base Fill Type

Similar to circle (see above).

## Anuloid - torus (pneumatika)

A doughnut-shaped primitive created by rotating a circle around an axis. The overall dimensions are defined by the Major and Minor Radius. The number of vertices (in segments) can be different for the circles and is specified in the context panel in the Tool Shelf with both radii (Major Segments and Minor Segments).

### Major Radius

Radius from the origin to the center of the cross sections

### Minor Radius

Radius of the torus's cross section

### Major Segments

Number of segments for the main ring of the torus. If you think of a torus as a "spin" operation around an axis, this is how many

steps in the spin.

Minor segments

Number of segments for the minor ring of the torus. This is the number of vertices of each circular segment.

Use Int+Ext Controls

Change the way the torus is defined:

Exterior Radius

When Use Int+Ext Controls is active, if viewed along the major axis, this is the radius from the center to the outer edge.

Interior Radius

When Use Int+Ext Controls is active, if viewed along the major axis, this is the radius of the hole in the center.

## Mříž

A standard grid is made out of  $n$  by  $m$  vertices. The resolution of the x-axis and y-axis can be specified in the context panel in the Tool Shelf which appears when the object is created; the higher the resolution, the more vertices are created. Example objects that can be created out of grids include landscapes (with the proportional editing tool or Displace modifier) and other organic surfaces. You can also obtain a grid when you create a plane and then use a subdivided modifier in Edit mode. However, there is a Landscape add-on available in the User Preferences.

X Subdivisions

The number of spans in the x direction. Minimum of 3, creating two face loops.

Y Subdivisions

The number of spans in the y direction.

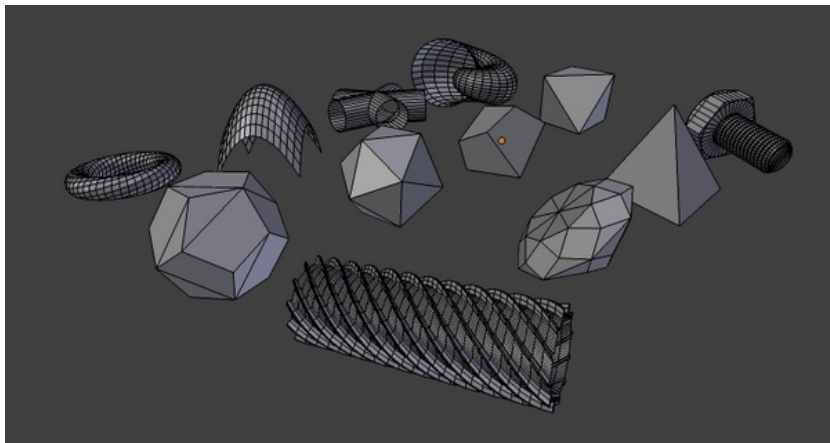
Size

The length of the sides of the grid.

## Opička

This is a gift from old NaN to the community and is seen as a programmer's joke or "Easter Egg". It creates a monkey's head once you press the Monkey button. The Monkey's name is "Suzanne" and is Blender's mascot. Suzanne is very useful as a standard test mesh, much like the [Utah Tea Pot](#) or the [Stanford Bunny](#).

## Doplňky



A few of the mesh primitives available as add-ons.

In addition to the basic geometric primitives, Blender has a constantly increasing number of script generated meshes to offer as pre-installed add-ons. These become available when enabled in the User Preferences' Add-ons section (filter by Add Mesh). Only a few are mentioned here:

### [Landscape](#)

Adds a landscape primitive. Many parameters and filters appear in the Tool Shelf.

### [Pipe Joints](#)

Adds one of five different pipe joint primitives. Radius, angle, and other parameters can be changed in the Tool Shelf.

### [Gears](#)

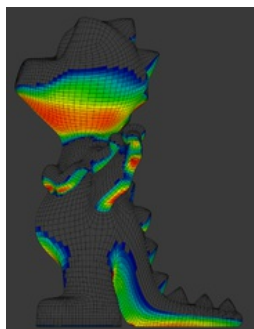
Adds a gear or a [worm](#) with many parameters to control the shape in the Tool Shelf.

## Analýza síťového modelu

analýza sítě (mesh analysis) je užitečný nástroj pro zobrazení atributů sítě, které mohou mít význam v určitém případě využití. Analýza sítě pracuje v editačním režimu objektu a vyznačuje oblasti pomocí barvy. Červené odpovídají vysokým hodnotám, modré nízkým a šedé oblasti značí hodnoty mimo rozsah.

Currently the different modes target 3d-printing as their primary use.

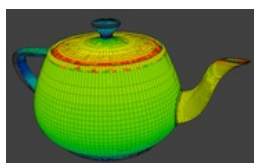
### Overhang



Overhang

Extrusion 3D printers have a physical limit to the overhang that can be printed, this display mode shows the overhang with angle range and axis selection.

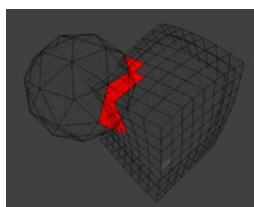
### Thickness



Thickness

Printers have a limited *wall-thickness* where very thin areas can't be printed, this test uses ray casting and a distance range to the thickness of the geometry.

### Intersections

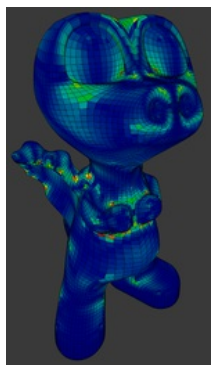


Intersecting faces

Another common cause of problems for printing are intersections between surfaces, where the inside/outside of a model can't be reliably detected.

Unlike other display modes, intersections have no variance and are either on or off.

### Distortion

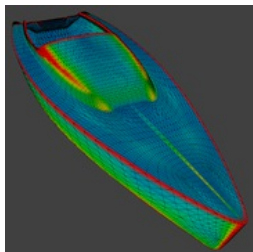


Distorted Faces

Distorted geometry can cause problems since the triangulation of a distorted ngon is undefined.

Distortion is measured by faces which are not flat, with parts of the face pointing in different directions.

### Sharp Edges



Sharp edges

Similar to wall-thickness, sharp edges can form shapes that are too thin to be able to print.

#### Warning



There are some known limitations with mesh analysis

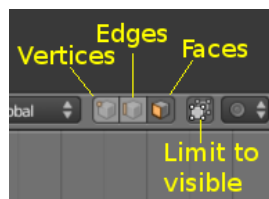
- Currently only displayed with deform modifiers.
- For high-poly meshes is slow to use while editing the mesh.

## Výběr prvků sítě

There are many ways to select elements, and it depends on what Mesh Select Mode you are in as to what selection tools are available. First we will go through these modes and after that a look is taken at basic selection tools.

## Režimy výběru

### Select Mode Header Widgets



Edit mode selection buttons

In Edit mode there are three different selection modes. You can enter the different modes by selecting one of the three buttons in the toolbar.

Using the buttons you can also use more than one selection mode at a time by ⇧ Shift LMB  clicking the buttons.

#### Vertices

Selected vertices are drawn in orange, unselected vertices in black, and the active or last selected vertex in white.

#### Edges

In this mode the vertices are not drawn. Instead the selected edges are drawn in orange, unselected edges black, and the active or last selected edge in white.

#### Faces

In this mode the faces are drawn with a selection point in the middle which is used for selecting a face. Selected faces and their selection point are drawn in orange, unselected faces are drawn in black, and the active or last selected face is highlighted in white.

Almost all modification tools are available in all three mesh selection modes. So you can Rotate, Scale, Extrude, etc. in all modes. Of course rotating and scaling a *single* vertex will not do anything useful without setting the pivot point to another location, so some tools are more or less applicable in some modes.

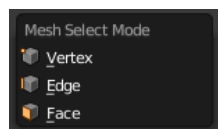
#### Note

The three selection mode buttons are only visible in Edit mode. The colors of selected, unselected and active components depend entirely on the current theme. Black, orange and white are from the default theme.

### Select Mode Pop-up

Mode: Edit mode

Hotkey: Ctrl⇧ Tab



Mesh Select Mode menu

You can also choose a selection mode with the pop-up menu

#### Select Mode » Vertices

Press Ctrl⇧ Tab and select Vertices from the pop-up menu, or press Ctrl⇧ Tab1.

#### Select Mode » Edges

Press Ctrl⇧ Tab and select Edges from the pop-up menu, or press Ctrl⇧ Tab2.

#### Select Mode » Faces

Press Ctrl⇧ Tab and select Faces from the pop-up menu, or press Ctrl⇧ Tab3.

### Switching select mode

When switching modes in an “ascendant” way (i.e. from simpler to more complex), from Vertices to Edges and from Edges to Faces, the selected parts will still be selected if they form a complete element in the new mode.

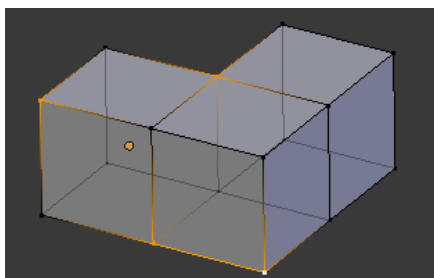
For example, if all four edges in a face are selected, switching from Edges mode to Faces mode will keep the face selected. All selected parts that do not form a complete set in the new mode will be unselected.

Hence, switching in a “descendant” way (i.e. from more complex to simpler), all elements defining the “high-level” element (like a face)

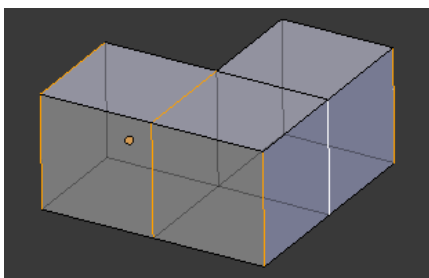
will be selected (the four vertices or edges of a quadrangle, for example).

By holding Ctrl when selecting a higher selection mode, all elements touching the current selection will be added, even if the selection does not form a complete higher element.

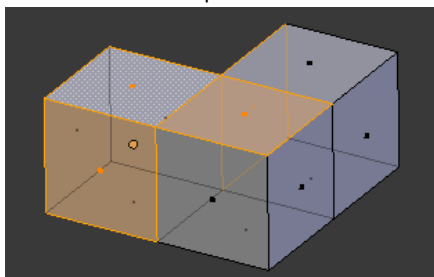
See (*Vertices mode example*), (*Edges mode example*), (*Faces mode example*) and (*Mixed mode example*) for examples of the different modes.



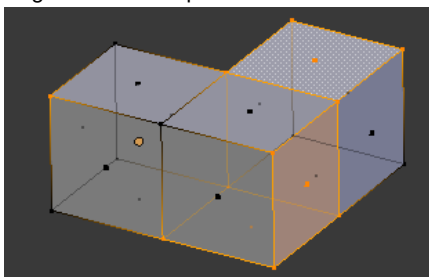
Vertices mode example.



Edges mode example.



Faces mode example.



Mixed mode example.

## Nástroje výběru

The select menu in edit mode contains tools for selecting components. These are described in more detail in the following pages.

### Border Select

Enables a rectangular region for selection

### Circle Select

Enables a circular shaped region for selection

### (De)select All A

Select all or none of the mesh components.

### Invert Selection CtrlI

Selects all components that are not selected, and deselect currently selected components.

### Select Random

Selects a random group of vertices, edges, or faces, based on a percentage value.

### Checker Deselect

Deselect alternating faces, to create a checker like pattern.

### Select Sharp Edges

This option will select all edges that are between two faces forming an angle less than a given value, which is asked you *via* a small pop-up dialog. The lower is this angle limit, the sharper will be the selected edges. At **180°**, **all** "manifold" (see below) edges will be selected.

### Linked Flat Faces (Ctrl⇧ ShiftAltF)

Select connected faces based on a threshold of the angle between them. This is useful for selecting faces that are planar.

### Interior Faces

Select faces where all edges have more than 2 faces.

### Side of Active

Selects all data on the mesh in a single axis

### Select Faces by Sides

Selects all faces that have a specified number of edges.

### Select Non Manifold (Ctrl⇧ ShiftAltM)

Selects vertices that are not completely bound by geometry, including border edges, floating edges, and orphan vertices. Only available in Vertex mode.

### Loose

Select all vertices or edges that do not form part of a face.

### Similar

Select components based on how similar certain properties are to it.

### More CtrlNum+

Propagates selection by adding components that are adjacent to selected elements.

### Less CtrlNum-

Deselects components that form the bounds of the current selection

### Mirror

Select mesh items at the mirrored location.

**Linked CtrlL**

Selects all components that are connected to the current selection.

**Vertex Path**

Selects a vertex path between two selected vertices

**Edge Loop**

Selects a loop of edges from a selected edge

**Edge Ring**

Selects edges parallel to a selected edge in the same ring of faces

**Loop Inner-Region**

Converts a closed selection of edges to the region of faces it encloses

**Boundary Loop**

Converts a selection of faces to the ring of edges enclosing it

## Vybíratelné prvky

Jak jsme viděli na stránce [struktura sítě](#), jsou sítě tvořené z různých typů prvků. Podle typu pohledu mají vizuální tvar rozličný, ale stále jde o reprezentaci základních prvků, tj. "vrcholů", "hran" a "stěn".

A tak je možné vybírat různé části i z těchto tří typů prvků. Zde je důležitá jedna suitečnost: pokud vybereme určitý typ prvku, (například hranu), tak implicitně jsou vybrány i její koncové vrcholy. Stejně tomu tak je v případě stěn. Toto je velice důležité, protože některé nástroje mohou operovat pouze vrcholy, hranami, plochami.

Obecně lze říci, že v určitou chvíli je možné vybrat pouze jeden element. Ale v případě použití různým výběrových režimů je možné dokonce i kombinovat výběry. Tj. například vybrat několik nesousedících vrcholů, hran a stěn. Tento kombinovaný výběrový režim je popsán dále.

## Režimy výběru

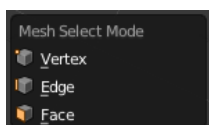
Jsou dva způsoby přepínání mezi výběrovými režimy:

### Select Mode popup

Mód: Edit mode

Klávesová zkratka: Ctrl⇧ Tab

V Edit režimu jsou k dispozici tři rozdílné výběrové režimy, viz (*Select Mode menu*).



Select Mode menu.

Select Mode » Vertices - vrcholy

Stiskněte Ctrl⇧ Tab a vyberte z popup menu Vertices, nebo stiskněte Ctrl⇧ Tab1 NumPad. Vybrané vrcholy jsou vykresleny žlutou barvou a nevybrané růžovou.

Select Mode » Edges

Stiskněte Ctrl⇧ Tab a vyberte z popup menu Edges, nebo stiskněte Ctrl⇧ Tab2 NumPad. V tomto režimu nejsou vrcholy vykreslovány a namísto toho jsou vybrané hrany vykresleny žlutě a nevybrané černě.

Select Mode » Faces

Stiskněte Ctrl⇧ Tab a vyberte z popup menu Faces, nebo stiskněte Ctrl⇧ Tab3 NumPad. V tomto režimu jsou stěny vykresleny s výběrovým bodem uprostřed každé z nich. Vybrané stěny jsou vykresleny žlutě s oranžovým výběrovým bodem uprostřed. Nevybrané jsou vykresleny černě.

Většina všech nástrojů uprav pracuje se všemi režimy výběru. A tak je možné rotovat, škálovat, Extrude - vytahovat, atd. ve všech režimech. Samotřejmě rotace *samotného* vrcholu není moc k užítu, pokud je střed otáčení nastaven právě v tomto vrcholu.

### Select Mode header widgets

Mode: Edit mode

Panel: Header of the 3D View



Edit mode select mode buttons.

You can also enter the different modes by selecting one of the three buttons in the toolbar; see (*Edit mode select buttons*).

Using the buttons you can also enter "mixed" or "combined" mode by ⇧ Shift LMB clicking the buttons. This will allow you to select vertices, edges and/or faces at the same time!

#### Note

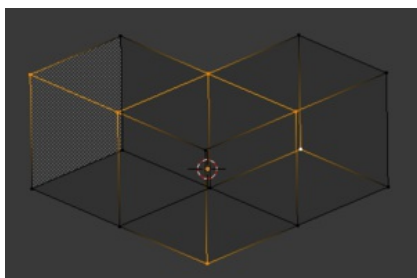
The "Mode Selection" buttons are only visible for meshes in Edit mode.

## Výběr prvků po přepnutí režimu výběru

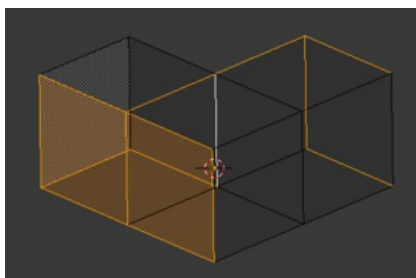
When switching modes in an "ascendant" way (i.e. from simpler to more complex), from Vertices to Edges and from Edges to Faces, the selected parts will still be selected if they form a complete set in the new mode. For example, if all four edges in a face are selected, switching from Edges mode to Faces mode will keep the face selected. All selected parts that do not form a complete set in the new mode will be unselected.

Hence, switching in a "descendant" way (i.e. from more complex to simpler), all elements defining the "high-level" element (like a face) will be selected (the four vertices or edges of a quadrangle, for example).

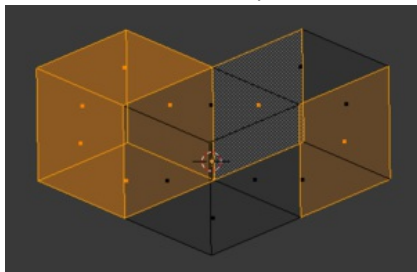
See (*Vertices mode example*), (*Edges mode example*), (*Faces mode example*) and (*Mixed mode example*) for examples of the different modes.



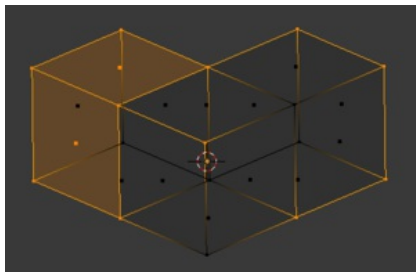
none Vertices mode example.



Edges mode example.



Faces mode example.




Mixed mode example.

## Základy výběrů

Mód: Edit režim

Klávesová zkratka: RMB  a ⇧ Shift RMB 

Nejrozšířeným způsobem pro výběr elementu je kliknutí RMB  na požadovanou položku. Tím je položka vybrána na úkor ostatních již vybraných.

## Přidávání do výběru

Pro přidání k existujícímu výběru podržíme ⇧ Shift a klikneme pravým tlačítkem myši. Dalším kliknutím bude vybraná komponenta z výběru odebrána ("odvybrána").

Stejně jako v objektovém režimu je *aktivní* unikátní prvek zobrazen světlejším odstínem (obecně poslední vybraný element). V závislosti na dále použitém nástroji může být tento výběr posledního prvku velice důležitý!

Za poznámku stojí skutečnost, že neexistuje volba pro výběr prvku, který je zakrytý ostatními prvky. Avšak pokud jsme v režimu pevného, stínovaného nebo texturového pohledu (nikoli obklopující hranoly, nebo drátový pohled), bude vybrána pouze viditelná část prvků. Tedy v případě krychle a bočního pohledu pouze viditelných například šest vrcholů, nikoli další.

## Výběr prvků v regionu

Mód: Edit mode

Klávesová zkratka: B, C, a Ctrl LMB  kliknout a táhnout

Vybírání v regionu umožňuje vybrat skupinu prvků uvnitř oblasti 2D a to přímo v podleh 3D. Regionem může být kružnice, nebo obdélník. Kružnicový region je dostupný pouze v Editačním režimu. Obdélníkový výběr nebo "*Výběr obrysu - Border Select*" je dostupný v obou režimech Editačním i Objektovém.


### Poznámka

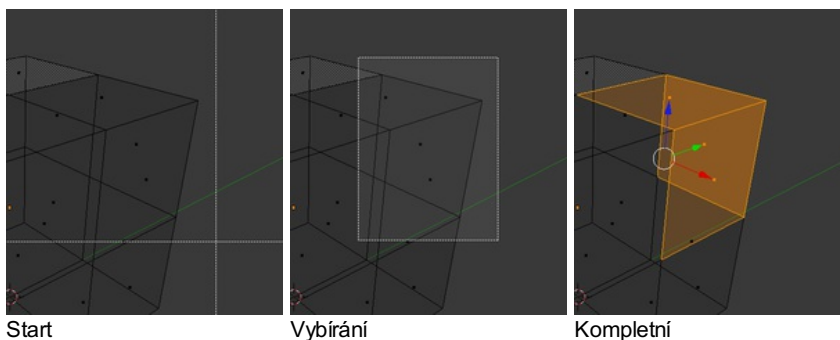
Vybrané prvky použitím obou metod je ovlivněno vlastností Limit Selection to visible (omezení výběru na viditelné). Vlastnost je dostupná v pohledu 3D v nastavení Solid Viewport Shading Mode.



Například:

1. v pevném režimu stínování a výběru povrchů budou všechny pochy *uvnitř* oblasti výběru vybrány;
2. Ve drátovém režimu pohledu budou vybrány pouze povrchy, jejichž střed spadá do oblasti výběru.


## Obdélníková oblast (výběr pozadí)

Funkce Border Select je dostupná v Edit módu nebo Object módu. Aktivaci nástroje provedeme pomocí B. Požití nástroje výběru Border Select k výběru skupiny objektů je takové, že kreslíme obdélník za současného podržení levého tlačítka myši LMB . In doing this you will select all objects that lie within or touch this rectangle. If any object that was last active appears in the group it will become selected *and* active.





V okamžiku (*Start*), Border Select je výběr aktivován a je indikován tečkovaným křížovým kurzorem. V kroku In (*Vybírání*) je opovídající oblast vybrána nakreslením obdélníku pomocí myši LMB . Oblast výběru je pokryta třemi stěnami. Uvolněním tlačítka myši LMB  je výběr kompletní.

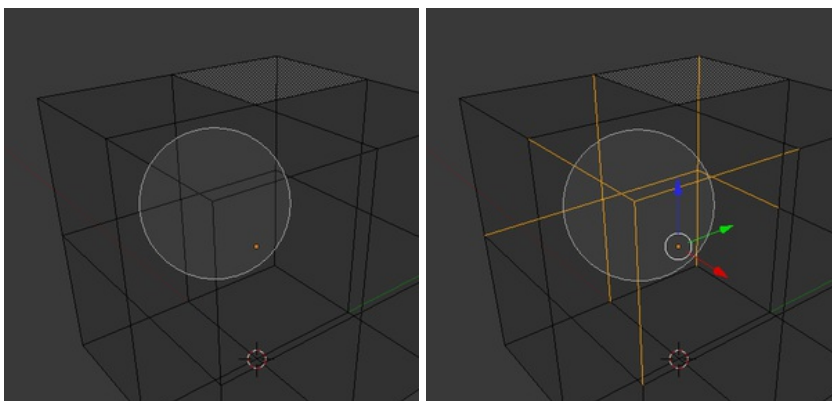
### Poznámka

Výběr pomocí Border přidává prvky k předchozímu výběru. Při začínání s výběrem je vhodné odznačit vše A a nakonec využít MMB  během kreslení hraničního obdélníku. Tímto krokem jsou z celkového výběru odebrány objekty uvnitř nakresleného hraničního obdélníku.

## Kruhový region

This selection tool is only available in Edit mode and can be activated with C. Once in this mode the cursor changes to a dashed cross-hair with a 2D circle surrounding it. The tool will operate on whatever the current select mode is. Clicking or dragging with the LMB , causing elements to be inside the circle will cause those elements to be selected.


Je možné zvětšit, nebo zmenšit region kruhu pomocí šedých kláves + NumPad a - NumPad, nebo točením středního myšního kolečka Wheel .





Před

Po

Výběr kruhového regionu

(*Výběr kruhového regionu*) je příkladem výběru hran v režimu Výběr hran. Jakmile hrana protíná kruh výběru, stává se vybranou. Nástroj je interaktivní kdy dochází k postupnému výběru hran zatímco kruhovým regionem pohybujeme se stisknutým LMB .


Pokud chcete vyjmout prvek z celkového výběru (tj. deselect), podržte MMB  nebo Alt LMB  a začněte kliknutím tahat znovu.

Pro výběrový režim povrchů (Faces) musí kružnice protínat indikátor plochy. Ten je patrný jako malý pixelový čtvereček uprostřed každé plochy.

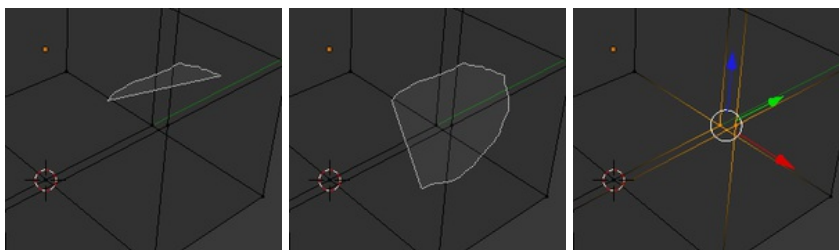
Pro ukončení tohoto nástroje postačuje kliknutí RMB  nebo stiskem klávesy Esc.

## Region laso

Nástroj výběru Lasso (česky překvapivě "laso") je podobný k výběru pomocí Border. Vybírá, které objekty jsou v regionu pouze s tím rozdílem, že Lasso je právě nakreslená oblast obecného tvaru..

Lasso je dostupné jak v režimu Edit Mode, tak i Object Mode. K aktivaci nástroje použijte Ctrl LMB  zatímco kreslíte. Jediným rozdílem mezi lasem a Border je v objektovém režimu (Object mode), kdy laso vybírá objekty jejichž střed spadá do požadovaného regionu.

Pro odvýběr použijte Ctrl+ Shift LMB  zatímco kreslíte.



Start

Vybírání

Kompletní

Výběr lasem

(*Výběr lasem*) je příklad použití nástroje výběru Lasso při zapnutém způsobu vybírání vrcholů (Vertex Select Mode).

## Doplňkové nástroje výběru

The select menu in edit mode contains additional tool for vybírání komponent:

(De)select All A

Vybrat všechny nebo odvybrat všechny síťové komponenty.

Invert Selection CtrlI

Inverze výběru. Vybere nevybrané a nevybere již vybrané komponenty.

More CtrlNum+

Propaguje výběr přidáním komponenty, která je nadřazená k vybraným elementům.

Less CtrlNum-

Odvýbere komponenty které ohraničují aktuální selekci.

## Rozšířené výběry

Kapitola rozšířených způsobů výběrů popisuje složitější způsoby výběrů jednotlivých komponent sítěového modelu. Odměnou za zvládnutí zdánlivé složitosti (pamatování si například horkých kláves) je možnost velice efektivně pracovat se sítěvými modely, zjednodušit si činnost výběru s následnou editací a pod. V mnohých instruktážních videích na internetu jsou obvykle demonstrátory tyto výběry využívány poměrně často a předpokládá se tedy jejich obecné technické zvládnutí. Tak, jako v jiných oblastech je samozřejmě zdůrazněna trpělivost a pila a zejména čas uživatele doby používání těchto technik.

Menu v režimu úprav obsahuje další nástroj pro výběr komponent:

Mirror - zrcadlo	vybere prvky sítě v zrcadleném prostoru.
Linked - linkované CtrlL	vybere komponenty připojené k vybraným.
Select Random náhodný výběr	Vybere náhodnou skupinu vrcholů, hran nebo stěn na základě procentuelní hodnoty.
Checker Deselect kostkovaný odvýběr	Odvybere stěny tak, že vznikne kostkovaný vzorek uspořádání výběru (šachovnice).
Select Every N Number of Vertices	Vybere vrcholy podle násobku N (tedy např. každý pátý).
Select Sharp Edges vybrat ostré hrany	Tato volba vybere všechny hrany mezi dvěma stěnami, které svírají úhel menší nežli zadaná hodnota (vstup dialogem). Čím je nižší tento limitní úhel, tím budou vybrány "ostřejší" hrany. V případě hodnoty <b>180°</b> budou vybrány <b>všechny</b> "typu manifold" (viz níže).
Linked Flat Faces Ctrl⇧	Vybere připojené plochy, jejichž sklon je menší než definovaná mez. Toto je užitečné pro výběr ploch, které jsou téměř v jedné rovině.
ShiftAltF Select Non Manifold Ctrl⇧ ShiftAltM	Vybírá vrcholy, které nejsou kompletně ohraničené geometricky včetně okrajových hran, plovoucích hran, visících vrcholů. Dostupné pouze v režimu výběru vrcholů a hran (Vertex a Edge).
Interior Faces vnitřní plochy Side of Active Aktivní stranu	Vybere stěny, jejichž všechny hrany mají více nežli dvě stěny. Tj. "vnitřní" stěny ve smyslu nikoli objemu, ale plochy.
Select Faces by Sides Plochy podle stran	Vybere všechna data sítě v jedné ose.
Loose Geometry geometricky ztracené	Vybere vrcholy nebo hrany které nejsou součástí povrchu. Tedy "ztracené prvky".

## Výběr podobných

Mód: Edit mode

Klávesová zkratka: ⇧ ShiftG

Menu: Select » Similar...

Vybírá komponenty, které mají podobné atributy jako vybraná, a to na základě hraniční meze (parametru), která je definována ve vlastnostech po aktivaci tohoto nástroje. Nástroje volby závisí na zvolené metodě vybírání

### Režim výběru vrcholů

Normal	Vybírá všechny vrcholy jejichž normály směřují podobně, jako vybraná komponenta.
Množství souvisejících stěn	Vybírá vrcholy, které mají k sobě připojený stejný počet stěn.
Skupiny vrcholů	Vybírá všechny vrcholy ze stejné <a href="#">skupiny vrcholů</a> .
Množství připojených hran	Vybírá všechny vrcholy které have mají k sobě připojený stejný počet hran.

### Režim výběru hran

Length - délka	Vybírá všechny hrany jejichž délka je podobná délce vybrané hrany.
Direction	Selects all edges that have a similar direction (angle) as those already selected.
Amount of Faces Around an Edge	Selects all edges that belong to the same number of faces.
Face Angles	Selects all edges that are between two faces forming a similar angle, as with those already selected.
Crease	Selects all edges that have a similar Crease value as those already selected. The Crease value is a setting used by the <a href="#">Subsurf Modifier</a> .
Bevel	Selects all edges that have the same Bevel Weight as those already selected.
Seam	Selects all edges that have the same Seam state as those already selected. Seam is a true/false setting used in <a href="#">UV-texturing</a> .
Sharpness	Selects all edges that have the same Sharp state as those already selected. Sharp is a true/false setting (a flag) used by the <a href="#">EdgeSplit Modifier</a> .

#### Režim výběru stěn

Material	Selects all faces that use the same material as those already selected.
Image	Selects all faces that use the same UV-texture as those already selected (see <a href="#">UV-texturing</a> pages).
Area	Selects all faces that have a similar area as those already selected.
Polygon Sides	Selects all faces that have the same number of edges.
Perimeter	Selects all faces that have a similar perimeter as those already selected.
Normal	Selects all faces that have a similar normal as those selected. This is a way to select faces that have the same orientation (angle).
Co-planar	Selects all faces that are (nearly) in the same plane as those selected.

## Vybírání smyček

You can easily select loops of components:

### Smyčky hran a vrcholů

Mód: Edit mode → Vertex nebo Edge select mode

Klávesová zkratka: Alt RMB  nebo CtrlE → Edge Loop

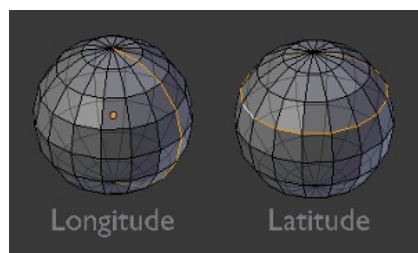
Menu: Select » Edge Loop nebo Mesh » Edges » Edge Loop

Holding Alt while selecting an edge selects a loop of edges that are connected in a line end to end, passing through the edge under the mouse pointer. Holding Alt+ Shift while clicking adds to the current selection.

Edge loops can also be selected based on an existing edge selection, using either Select » Edge Loop, or the Edge Loop Select option of the Edge Specials menu (CtrlE).

Vertex mode

In Vertex select mode, you can also select edge loops, by using the same hotkeys, *and clicking on the edges* (not on the vertices).



Longitudinal and latitudinal edge loops.

The left sphere shows an edge that was selected longitudinally. Notice how the loop is open. This is because the algorithm hit the vertices at the poles and terminated because the vertices at the pole connect to more than four edges. However, the right sphere shows an edge that was selected latitudinally and has formed a closed loop. This is because the algorithm hit the first edge that it started with.

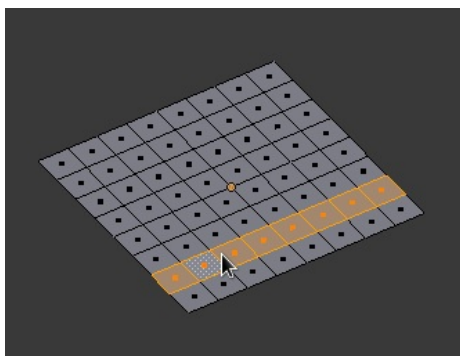
### Face Loops

Mód: Edit mode → Face or Vertex select modes


Klávesová zkratka: Alt RMB 

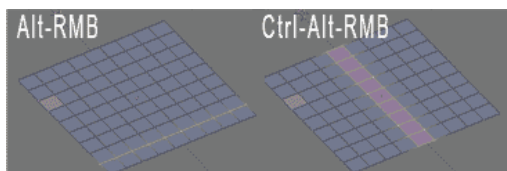
In face select mode, holding Alt while selecting an **edge** selects a loop of faces that are connected in a line end to end, along their opposite edges.

In vertex select mode, the same can be accomplished by using CtrlAlt to select an edge, which selects the face loop implicitly.




Face loop selection.

This face loop was selected by clicking with Alt RMB  on an edge, in face select mode. The loop extends perpendicular from the edge that was selected.




Alt versus CtrlAlt in vertex select mode.

A face loop can also be selected in Vertex select mode. Technically CtrlAlt RMB  will select an Edge Ring, however in Vertex select mode, selecting an Edge Ring implicitly selects a Face Loop since selecting opposite edges of a face implicitly selects the entire face.

## Edge Ring

Mode: Edit mode → Edge select mode

Hotkey: CtrlAlt RMB  or CtrlE → Select » Edge Ring

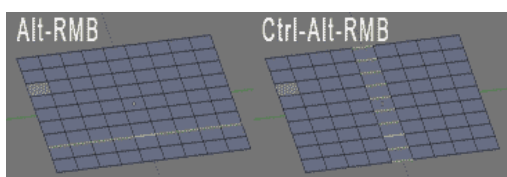
Menu: Select » Edge Ring or Mesh » Edges » Edge Ring

In Edge select mode, holding CtrlAlt while selecting an edge selects a sequence of edges that are not connected, but on opposite sides to each other continuing along a [face loop](#).

As with edge loops, you can also select edge rings based on current selection, using either Select » Edge Ring, or the Edge Ring Select option of the Edge Specials menu (CtrlE).

Vertex mode

In Vertex select mode, you can use the same hotkeys when *clicking on the edges* (not on the vertices), but this will directly select the corresponding face loop...



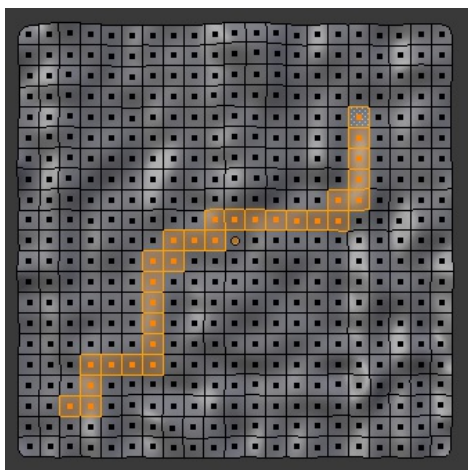
A selected edge loop, and a selected edge ring.

In (A *selected edge loop*, and a *selected edge ring*), the same edge was clicked on but two different “groups of edges” were selected, based on the different commands. One is based on edges during computation and the other is based on faces.

## Path Selection

Mode: Edit mode

Hotkey: Ctrl RMB  and the menu item Select → Shortest Path



Select a face or vertex path with Ctrl RMB 

Selects all geometry along the shortest path from the active vertex/edge/face to the one which was selected.

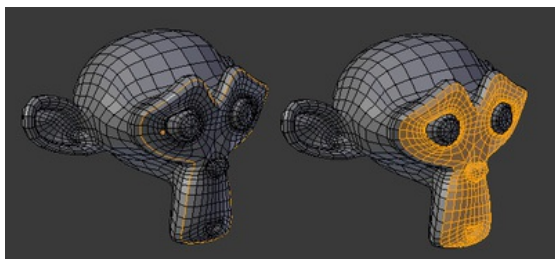
### Loop Inner-Region

Mode: Edit mode → Edge select mode

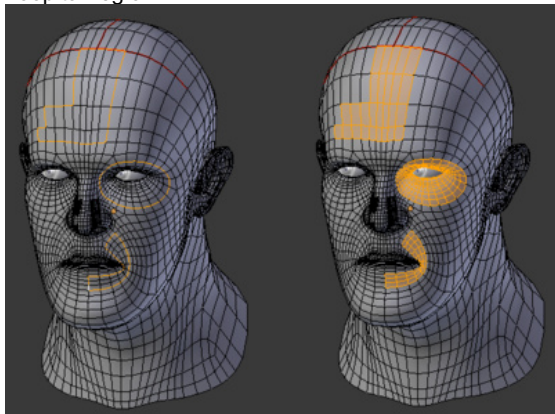
Hotkey: CtrlE → Select Loop Inner-Region

Menu: Select » Select Loop Inner-Region or Mesh » Edges » Select Loop Inner-Region

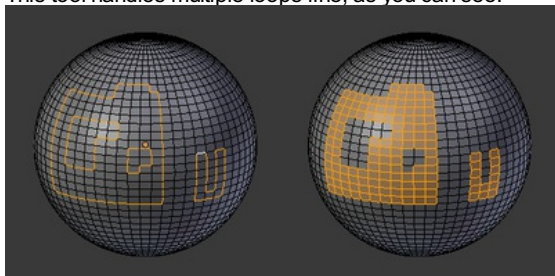
Select Loop Inner-Region selects all edges that are inside a closed loop of edges. While it is possible to use this operator in Vertex and Face selection modes, results may be unexpected. Note that if the selected loop of edges is not closed, then all connected edges on the mesh will be considered inside the loop.



Loop to Region.



This tool handles multiple loops fine, as you can see.



This tool handles "holes" just fine as well.

### Boundary Loop

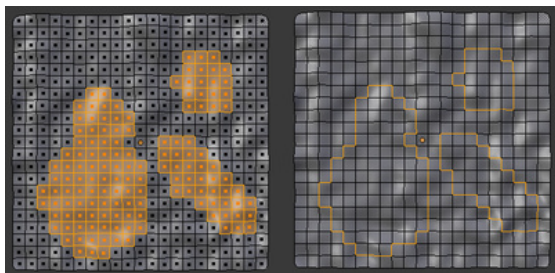
Mode: Edit mode → Edge select mode

Hotkey: CtrlE → Select Boundary Loop

Menu: Select » Select Boundary Loop or Mesh » Edges » Select Boundary Loop

Select Boundary Loop is the “logical inverse” of Select Loop Inner-Region, based on all regions currently selected, it selects only the edges at the border of these regions. It can operate in any select mode, but will always switch to Edge select mode when run.

All this is much more simple to illustrates with examples:



Select Boundary Loop does the opposite and forces into Edge Select Mode

## Výběr hran



Hrany mohou být vybírány podobně jako vrcholy a stěny - pravým kliknutím při zapnutém režimu výběru hran. Stiskem  $\diamond$  Shift během kliknutí dochází k přidání/odebrání hrany k výběru.

## Smyčky hran

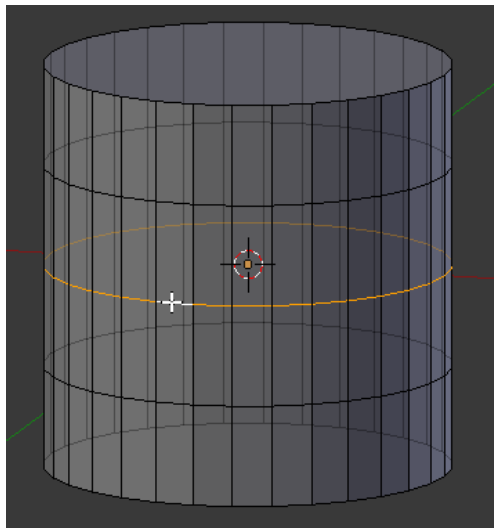
Mód: Edit Mode (Mesh)

Klávesová zkratka: Alt RMB - nebo  $\diamond$  ShiftAlt RMB pro úpravu existujícího výběru

Menu: Select » Edge Loop

Edge loops can be selected by first selecting an edge (vertex or edge selection mode), and then going to Select » Edge Loop. The shortcut Alt RMB on an edge (either vertex or edge select mode) is a quicker and more powerful way of doing so. More powerful, because you can add/remove loops from an existing selection if you press  $\diamond$  Shift too.

Note, that if you want to select a loop while being in vertex select mode, you still have to perform the shortcut on an edge - while you, for just selecting vertices, would rightclick on a vertex.



Alt on Linux

Alt is on some Linux distros caught by the windows manager. If you see the above shortcut not working, make sure that blender can properly recognize the usage of Alt.

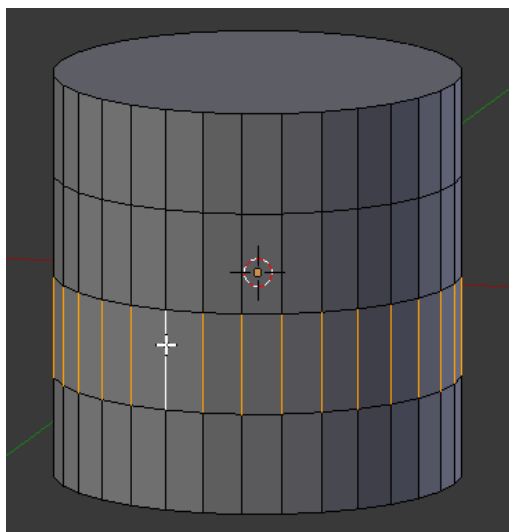
## Prstence hran

Mód: Edit Mode (Mesh)

Klávesová zkratka: AltCtrl RMB - nebo  $\diamond$  ShiftAltCtrl RMB pro úpravu existujícího výběru

Menu: Select » Edge Ring

Edge Rings are selected similarly. Based on the selection of an edge go to Select » Edge Ring. Or use AltCtrl RMB on an edge.



Convert selection to whole faces

If the edge ring selection happened in Edge Select Mode, switching to Face Select Mode will erase the selection.

This is because none of those faces had all its (four) edges selected, just two of them.

Instead of selecting the missing edges manually or by using  $\diamond$  ShiftAlt RMB twice, it is easier to first switch to Vertex Select Mode,

which will kind of "flood" the selection. A subsequent switch to Face Select Mode will then properly select the faces.

## Výběr stěn



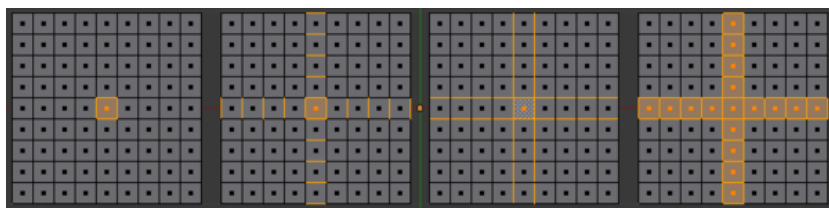
Pro výběr stěn modelu je nutné přejít do tohoto režimu pomocí tlačítka na obrázku vlevo, nebo stiskem kombinace **Ctrl+Tab** a výběrem z menu. Výběr obvykle pracuje s **RMB** ; pro přidání stěny k výběru. Pro odebrání z výběru před kliknutím stisknout **Shift**.

## Smyčky stěn

Mód: Edit Mode (Mesh)

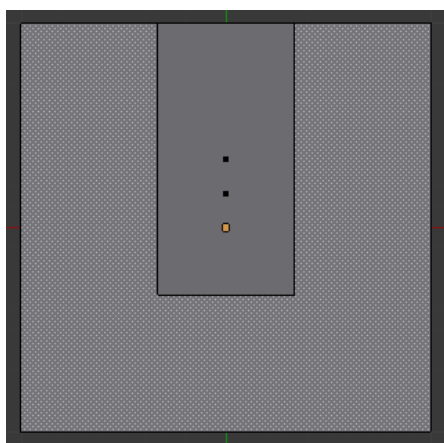
Klávesová zkratka: **Alt RMB** - nebo **ShiftAlt RMB** pro úpravu existujícího výběru

Face Loops are pretty much the same as Edge Rings. If you want to select a Face Loop, there is no menu entry that works based on a selected face. Using **Select » Edge Ring** would select a "cross" with the prior selected face as the middle. If you want to avoid switching to Edge Select Mode to select a Face Loop, use the **Alt RMB** shortcut.



1. Just the selected face.
2. Select the face, then **Select » Edge Ring**. See, how Blender selects edges, even if being in Face Select Mode. If these edges are desired and you want to work on them, switch in Edge Select Mode. Switching to Vertex Select Mode would flood the selection and leave you with the 4th image as result, after going back to Face Select Mode.
3. Select the face, the **Select » Edge Loop**. As in the example above, Blender pretends to be in Edge Select Mode and takes the four edges of the selected face as base for the selection operation.
4. This selection was created by **Alt RMB** on the left edge of the center face, followed by twice **ShiftAlt RMB** on the top edge of the center face. Two times, because the first click will remove the selected face loop (in this case, just the original selected face), while the second click will add the whole vertical running loop to the selection, creating the cross.

## N-úhelníky v režimu výběru stěn



As already known, faces are marked with a little square dot in the middle of the face. With ngons that can lead in certain cases to a confusing display. The example shows the center dot of the U-shaped ngon being inside of the oblong face inside the "U". It is not easy to say which dot belongs to which face (the orange dot in the image is the object center). Luckily, you don't need to care much - because to select a face, you don't have to click the center dot, but the face itself.



### Face selection

*To select a face:*

Click the face, not the dot!

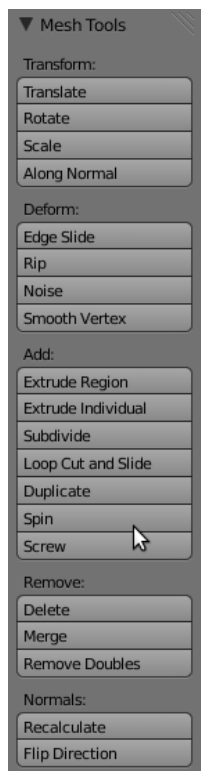
## Úpravy sítí

Blender nabízí uživatelům celou sadu nástrojů pro úpravy sítí modelů. Jsou dostupné přes menu Mesh Tools-nástroje sítě, ve 3D pohledu je to menu sítí-mesh, a kontextová nabídka ve 3D pohledu. Samozřejmě jsou podporovány horké klávesy pro nejpoužívanější funkce.

Note that all the “transform precision/snap” keys (Ctrl and/or ⇧ Shift) work also for all these advanced operations... However, most of them do not have [axis locking](#) possibilities, and some of them do not take into account [pivot point](#) and/or [transform orientation](#) neither...

These transform tools are available in the Transform section of the Mesh menu in the menu bar. Note that some of these can also be used on other editable objects, like curves, surfaces, and lattices.

## Typy nástrojů



### Nástroje sítí

K jednotlivým nástrojům pro úpravu sítí je možné se dostat více cestami.

#### [Transformační a deformační nástroje:](#)

- Posuv (Translate)
- Rotace
- Škálování (Scale)
- Zrcadlení (Mirror)
- Shrink/Flatten/Along Normal
- Push/Pull
- Do koule (To Sphere)
- Shear
- Warp
- Posuv hran (Edge Slide)
- Posun vrcholů (Vertex Slide)
- Šum (Noise)
- Uhlazení sítě (Smooth Vertex)
- Rotace hrany

#### [Přídávací a dělicí nástroje:](#)

- Vytvoření hrany/stěny
- Vyplnit
- Krásně vyplnit
- Vytvořit pevný objekt
- Čtyřúhelníky do trojúhelníků
- Extrude Region
- Extrude Individual
- Subdivide
- Loop Cut/Slide
- Knife tool
- Vertex connect
- Duplicate
- Spin
- Screw
- Symmetrize
- Inset
- Bevel
- Wireframe

#### [Spojovací a čistící nástroje:](#)

- Smazat (Delete)
- Dissolve
- Spojit (Merge)
- Automaticky spojit
- Odstranit duplicity
- Trojúhelníky do čtyřúhelníků

#### [Nástroje oddělování:](#)

- Rip
- Rip fill
- Split
- Separate
- Edge Split

## Přístup k nástrojům sítí

### Paleta nástrojů

Když vyberete síťovinu a vstoupíte pomocí ⇌ Tab do editačního režimu, pak se sada Tool Shelf změní z nástrojů objektu Object Tools na nástroje síťoviny (Mesh Tools). Toto jsou pouze některé z editačních nástrojů sítí.

### Menu

Síť (Mesh) je umístěno v liště záhlaví. Některé menu jsou přístupné pomocí klávesových zkratk:

CtrlF otevře menu nástrojů stěn (Face)

CtrlE otevře menu nástrojů hran (Edge)

CtrlV otevře menu nástrojů vrcholů (Vertex)

## Základy úpravy sítě

Mezi základní operace úprav sítě objektu patří **posuv**, **rotace**, **škálování**. Těmito třemi operacemi je obecně možné provést libovolnou změnu prostorového uspořádání vrcholů, hran a stěn tělesa reprezentovaného sítí.

Další operace jako například zrcadlení (škálování faktorem -1 podél osy, která je kolná na rovinu zrcadelní).. je možné popsat těmito základními operacemi (aplikované na celý objekt, či pouze na jeho část). Jednotlivé operace jsou popsány v následujících sekcích:

- [Posuv, rotace, měřítko](#)
- [Přidávání prvků](#)
- [Mazání prvků](#)
- [Vytváření stěn a hran](#)
- [Zrcadlení](#)

Posuv, rotace, měřítko

Mód: editační režim

Panel: Mesh Tools (Editing context)

Klávesová zkratka: G/R/S

Menu: Mesh » Transform » Grab/Move, Rotate, Scale, ...

Pokud vyberete jeden nebo více prvků, je možné je posunovat, rotovat (R) nebo škálovat (S) tak, jak je popsáno v sekci [manipulace v 3D prostoru](#).

**Poznámka:** Pojem měřítko použijeme ve statickém popisu skutečnosti, zatímco pojmem škálování chápeme operaci změny, tj. změny měřítka. Záměna obou slov je tedy relevantní.

Pro posun, rotaci a škálování vybraných prvků použijte volbu z menu Translate, Rotate, and Scale, což je také popsáno v kapitole [Tranformační manipulátory](#), anebo mnohem rychleji pomocí kláves:

G, R a S.

Po přesunu prvků je na panelu nástrojů k dispozici výpis detailů změn, omezovací efekt pro různé osy, vypnutí a zapnutí proporcionálního škálování atd.

Samozřejmě pokud přesunete prvek daného typu (například hranu), současně tím implikujeme změnu souvisejících prvků i například jiných typů (vrcholy, stěny).

Je možné též v editačním režimu využít rozšířenou volbu použití základních manipulátorů, jak je popsáno v [proporcionální editaci](#).

Přidávání objektů

Anglicky není vůbec hotové, dal jsem návrh. Zde je překlad návrhu:

This page is being developed right now, follow this link to see the page while it's being constructed.

Page status ([reviewing guidelines](#))

**Void page**

**Proposed fixes:** [X](#)

Object mode Adding object - menu, manual selection - is under last added object... or shortcut ⇧ ShiftA.

Edit mode: adding object -> addin mesh, e.g. final merging into one object.

center for adding: X 3D curson. If you need to center cursor imed. just press ⇧ ShiftC, or edit in the N tools menu cursor xordinates x,y,z. Content proposal from [Quark66](#) 12:53, 11 June 2013 (CEST):

## Mazání a spojování

Pro odstranění komponenty z prostoru je možné využít několik způsobů, od nejjednodušších (mazání), až například pro odstraňování jednotlivých prvků sítě v editačním režimu. Mazání a související spojování je závislé na kontextu použití a je popsáno dále.

## Smazat (Delete)

Delete (X nebo Del)

smaže vybrané vrcholy, hrany nebo stěny. Tato operace může být také omezena na:

Vertices

Vymaže všechny vrcholy ve vybrané oblasti. Odstraní také stěny nebo hrany, které jsou k nim připojeny.

Edges

Vymaže všechny hrany ve vybrané oblasti. Odstraní také stěny, které jsou k nim připojeny.

Faces

Vymaže stěny ve vybrané oblasti.

Only Edges & Faces

Omezí operaci pouze na vybrané hrany a související stěny.

Only faces

Odstraní pouze stěny, ale hrany zůstávají zachovány.

Dissolve

Odstraní vybranou geometrii, ale ponechá povrch uzavřený, efektivně přemění oblast výběru do jednoho polygonu (N-gon). Odstranění je malinko odlišné od toho, když máme vybrané vrcholy, hrany a stěny. Můžeme přidat dodatečné detaily, nebo zkrátka rychle odmazat požadované.

Limited Dissolve

Omezené odmazání pracuje na rovinných stěnách a na hranách s nastavitelným prahem sousedících úhlů.

Edge Collapse

Zhroutí hrany do jednoho vrcholu. Často využíváno v ostranění smyčky stěn.

Edge Loop

Maže smyčky hran. Pokud není vybraná oblast smyčka hran, operace je bez efektu..

## Konverze trojúhelníků na čtyřúhelníky

Tris to Quads Alt+J Vezme sousedící triangly a odstraní jejich sdílenou hranu tak, že vznikne čtyřúhelník. Nástroj může být využitý i pro více trojúhelníků.

Stejný výsledek získáme při výběru dvou trojúhelníků a použití zkratky F pro vytvoření stěny.

## Propojení, slití (Unsubdivide)

Mode: editační režim

Hotkey: Ctrl+E » Unsubdivide

Menu: Mesh » Edges » Unsubdivide

Funkce Unsubdivide je opačná k funkci subdivide (členění) a pokouší se odstranit hrany, které vznikly jako výsledek dřívější operace dalšího členění. Pokud po operaci členění došlo k dodatečné editaci, mohou se dostavit neočekávané výsledky.

Iterations

Iterací. Kolik podčlenění má být odstraněno.

## Spojování

### Spojování vrcholů

Mode: Edit mode

Hotkey: Alt+M

Menu: Mesh » Vertices » Merge..., Specials » Merge or Vertex Specials » Merge

This tool allows you to merge all selected vertices into an unique one, deleting all others. You can choose the location of the surviving vertex in the menu this tool pops-up before executing:

At First

Only available in Vertex select mode, it will place the remaining vertex at the location of the first one selected.

#### At Last

Only available in Vertex select mode, it will place the remaining vertex at the location of the last one selected (the active one).

#### At Center

Available in all select modes, it will place the remaining vertex at the center of the selection.

#### At Cursor

Available in all select modes, it will place the remaining vertex at the 3D Cursor.

#### Collapse

This is a special option, as it might let “alive” more than one vertex. In fact, you will have as many remaining vertices as you had “islands” of selection (i.e. groups of linked selected vertices). The remaining vertices will be positioned at the center of their respective “islands”. It is also available *via* the Mesh » Edges » Collapse menu option...

Merging vertices of course also deletes some edges and faces. But Blender will do everything it can to preserve edges and faces only partly involved in the reunion.

### AutoMerge Editing

Mode: Edit mode

Menu: Mesh » AutoMerge Editing

The Mesh menu as a related toggle option: AutoMerge Editing. When enabled, as soon as a vertex moves closer to another one than the Limit setting (Mesh Tools panel, see below), they are automatically merged.

### Remove Doubles

Mode: Edit mode

Panel: Editing context → Mesh Tools

Hotkey: W » 4 or CtrlV » Remove doubles

Menu: Mesh » Vertices » Remove Doubles, Specials » Remove Doubles or Vertex Specials » Remove Doubles

Remove Doubles is a useful tool to simplify a mesh by merging vertices that are closer than a specified distance to each other. An alternate way to simplify a mesh is to use the [Decimate modifier](#).

#### Merge Distance

Sets the distance threshold for merging vertices, in Blender units.

#### Unselected

Allows vertices in selection to be merged with unselected vertices. When disabled, selected vertices will only be merged with other selected ones.

## Vytvoření hrany/stěny

Mód: Edit mode

Klávesová zkratka: F

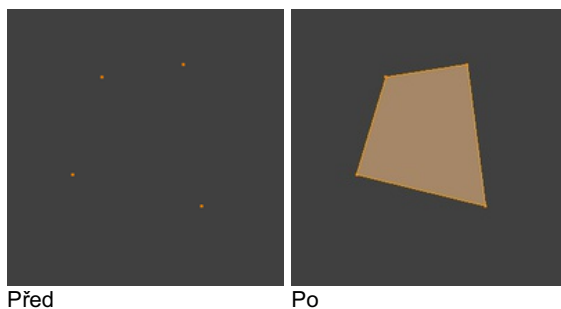
Menu: Mesh → Faces → Make Face/Edge

Toto je kontextově vázaný nástroj vytvářející geometrii výplní výběru. Pokud jsou vybrány 2 vrcholy, vytvoří se hrana. V opačném případě vytvoří stěny.

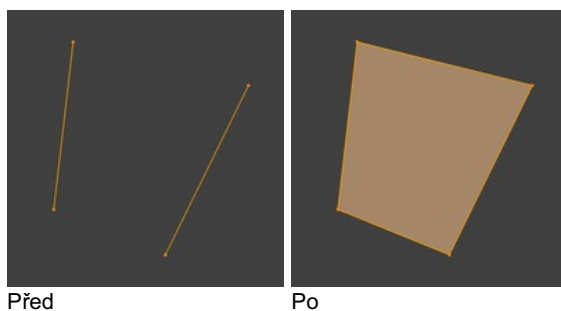
Typickým případem použití je výběr vrcholů a stisk F, avšak Blender také podporuje vytváření stěn z odlišných výběrů by pomohl rychle a vytvořit požadovanou geometrii.

Následující metody jsou použity automaticky v závislosti na kontextu.

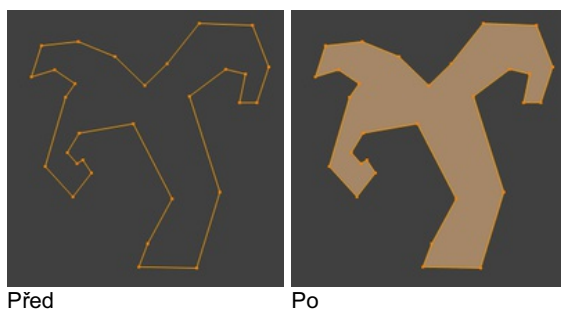
Izolované vrcholy.



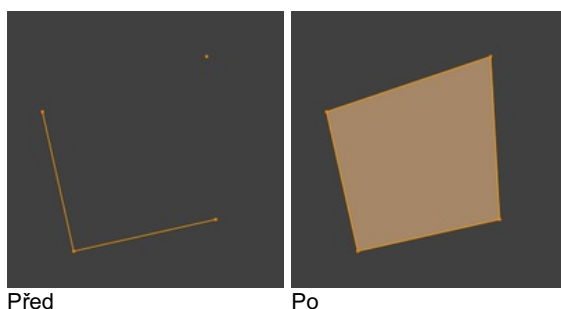
Izolované hrany.



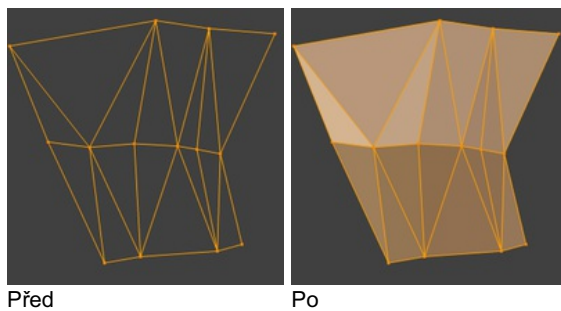
N-gon z hran: Pokud je vybráno více hran, Blender vytvoří polygon (ngon). Za poznámku stojí skutečnost, že Blender takto nevytvoří díry. Pro jejich vytváření je nutné použít nástroje povrchů [XXX Fill Faces tool](#).



Směs vrcholů a hran: existující hrany jsou použity pro vytvoření povrchů stejně jako extra vrcholy.



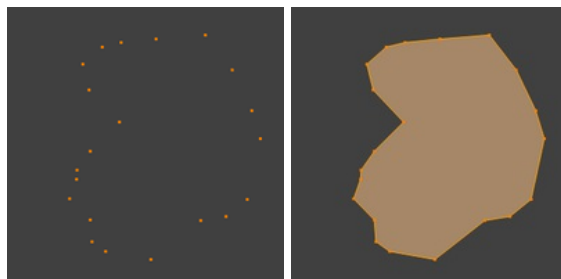
Hrany-sít: někdy můžete mít mnoho spojených hran bez vnitřních povrchů.



Před

Po

Mrak bodů: *pokud je mnoho izolovaných vrcholů, Blender vypočítá hrany pro polygon (n-gon).*



Před

Po

Výběr samostatných vrcholů: s výběrem samostatných vrcholů na okraji bude vytvořen povrch podél tohoto okraje. To ušetří manuální výběr dalších dvou vrcholů. Tento nástroj je možné využít vícenásobně jako pokračování tvorby povrchů.



#### Další odkazy

Pro další způsoby vytváření stěn viz:

- [Fill](#)
- [Grid Fill](#)
- [Bridge Edge Loops](#)

Zrcadlové editování

## Zrcadlení-X

Mód: editační režim

Panel: Mesh Options » X-mirror

volba X-mirror z panelu volby sítě Mesh Options nabízí možnost editace "obou stran" síťoviny naráz. Pokud transformujete element (vrchol, hranu, stěnu), je automaticky transformován i protějšek podle zrcadlení rovinou, která je kolmá na osu X a procházející (lokální prostor).

## Modifikátor zrcadla

Možnosti modifikátoru ve směru osy X jsou trochu omezenější, což může být důvodem k omezené použitelnosti. Naopak pokud máte již zpracovanou přesnou polovinu síťoviny objektu, je jednodušší využít tohoto modifikátoru. viz [Modifikátor zrcadlení](#)

## Symmetrizace sítě

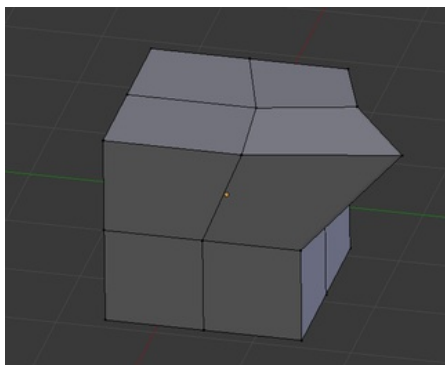
Mód: editační režim

Menu: Mesh » Symmetrize

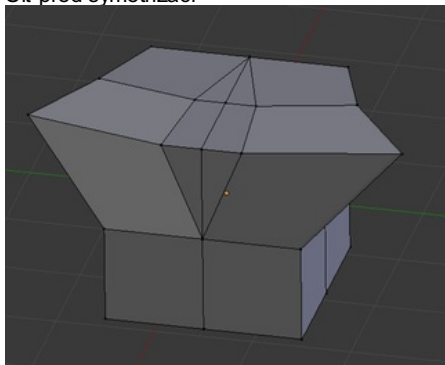
Nástroj symmetrizace dává jednoduchou cestu pro vytvoření symetrické sítě. symmetrizace pracuje tak, že ořízne síť v pivot bodě objektu a zrcadlí geometrii podél dané osy a výsledné dvě poloviny objektu spojí

Směr

Specifikuje osu a směr působení efektu. Může to být jedna ze tří os a to kladná, nebo záporná.



Síť před symetrizací



Síť po symetrizaci

## Zrcadlení geometrie

Viz kapitola [zrcadlo](#) pro další informace o využití zrcadlení.

Nástroje pro úpravy vrcholů

Tato kapitola popisuje celou řadu nástrojů dostupných v menu operace nad vrcholy Mesh » Vertices. Nástroje jsou určeny především pro vrcholy sítě modelu, nebo jejich skupiny, ale je možné je uplatnit i v práci s hranami, nebo stěnami.

## Spojování (Merging)

### Spojování vrcholů

Mód: editační režim

Klávesová zkratka: AltM

Menu: Mesh » Vertices » Merge..., Specials » Merge nebo Vertex Specials » Merge

Nástroj umožňuje spojovat všechny vybrané vrcholy do jednoho (posunem). Nástroj před spojením dá uživateli možnost několika způsobů spojení:

At First

Dostupné pouze v režimu výběru vrcholů Vertex. Umístí ostatní vrcholy na místo prvního v pořadí, v kterém byly vybírány.

At Last

Dostupné pouze v režimu výběru vrcholů Vertex. Umístí ostatní vrcholy na místo posledního v pořadí, v kterém byly vybírány.

At Center

Dostupné pouze v režimu výběru vrcholů Vertex. Umístí vrcholy na středové místo všech vybraných.

At Cursor

Dostupné pouze v režimu výběru vrcholů. Umístí vrcholy na pozici 3D kurzoru.

Collapse

This is a special option, as it might let "alive" more than one vertex. In fact, you will have as much remaining vertices as you had "islands" of selection (i.e. groups of linked selected vertices). The remaining vertices will be positioned at the center of their respective "islands". It is also available *via* the Mesh » Edges » Collapse menu option...

Merging vertices of course also deletes some edges and faces. But Blender will do everything it can to preserve edges and faces only partly involved in the reunion.

### Automatické spojování

Mód: editační režim

Menu: Mesh » AutoMerge Editing

The Mesh menu as a related toggle option: AutoMerge Editing. When enabled, as soon as a vertex moves closer to another one than the Limit setting (Mesh Tools panel, see below), they are automatically merged.

### Odstranění duplicit

Mód: editační režim

Panel: Editing context → Mesh Tools

Klávesová zkratka: W » 4 or CtrlV » Remove doubles

Menu: Mesh » Vertices » Remove Doubles, Specials » Remove Doubles or Vertex Specials » Remove Doubles

Remove Doubles is a useful tool to simplify a mesh by merging vertices that are closer than a specified distance to each other. An alternate way to simplify a mesh is to use the [Decimate modifier](#).

Merge Distance

Sets the distance threshold for merging vertices, in Blender units.

Unselected

Allows vertices in selection to be merged with unselected vertices. When disabled, selected vertices will only be merged with other selected ones.

## Oddělení

### Roztřžení

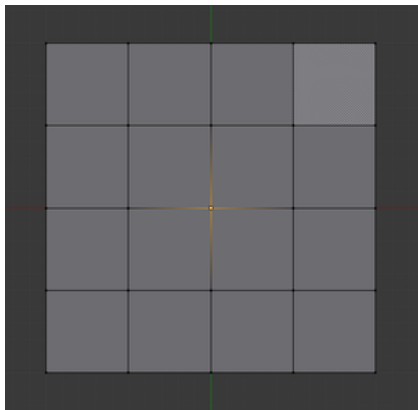
Mód: editační režim

Klávesová zkratka: V

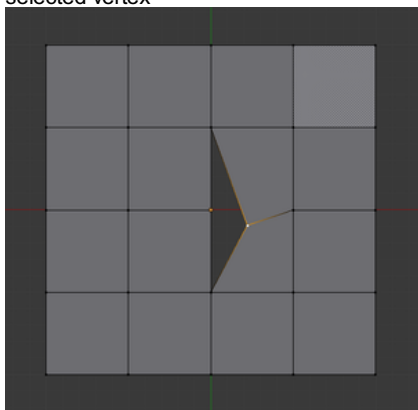
Menu: Mesh » Vertices » Rip

Rip creates a "hole" into a mesh by making a copy of selected vertices and edges, still linked to the neighbor non-selected vertices, so that the new edges are borders of the faces on one side, and the old ones, borders of the faces of the other side of the rip.

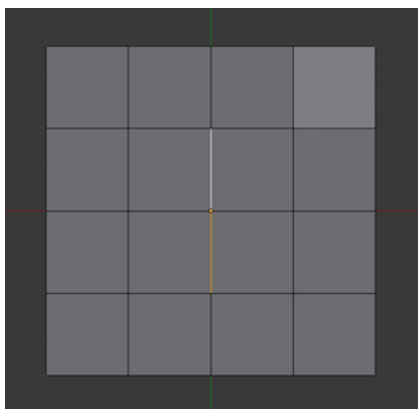
## Příklady



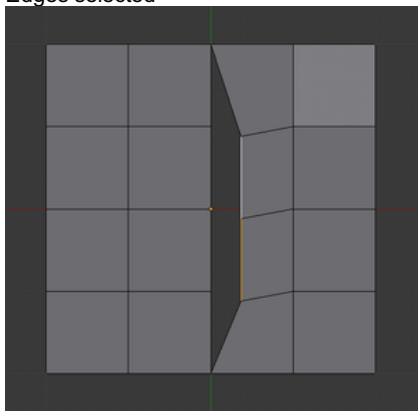
selected vertex



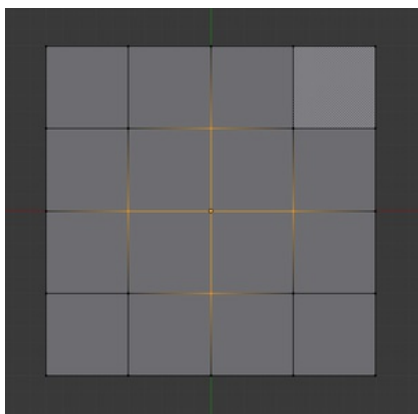
Hole created after using rip on vertex



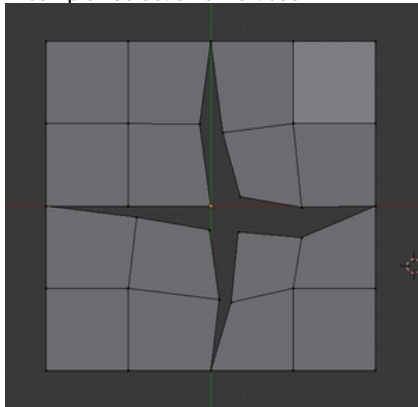
Edges selected



Result of rip with edge selection



A complex selection of vertices



Result of rip operation

## Omezení

Rip will only work when edges and/or vertices are selected. Using the tool when a face is selected (explicitly or implicitly), will return an error message "Can't perform ripping with faces selected this way". If your selection includes some edges or vertices that are not "between" two faces (manifold), it will also fail with message "No proper selection or faces include".

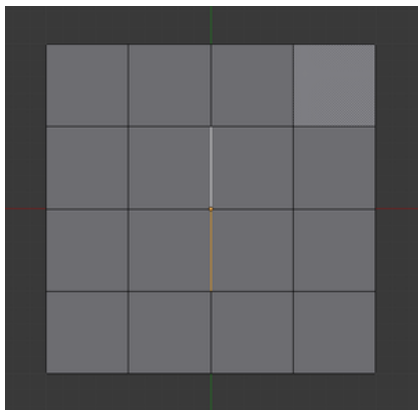
## Vyplněné roztřžení

Mód: editační režim

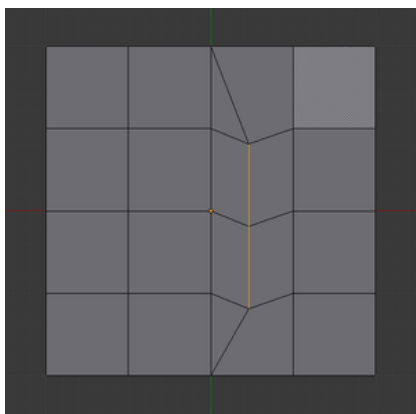
Klávesová zkratka: AltV

Menu: Mesh » Vertices » Rip Fill

Rip fill works the same as the Rip tool above, but instead of leaving a hole, it fills in the gap with geometry.



Edges selected



Result of rip fill

## Split

Mód: editační režim

Klávesová zkratka: Y

Menu: Mesh » Vertices » Split

A quite specific tool, it makes a sort of copy of the selection, removing the original data *if it is not used by any non-selected element*. This means that if you split an edge from a mesh, the original edge will still remain unless it is not linked to anything else. If you split a face, the original face itself will be deleted, but its edges and vertices remain unchanged. And so on.

Note that the “copy” is left exactly at the same position as the original, so you must move it (G) to see it clearly...

## Oddělení

Mód: editační režim

Klávesová zkratka: P

Menu: Mesh » Vertices » Separate

This will separate the selection in another mesh object, as described [here](#).

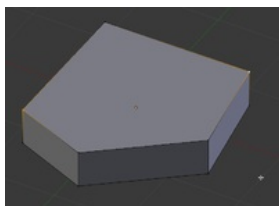
## Propojení vrcholů

Mód: editační režim

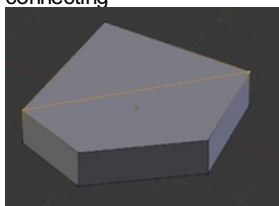
Klávesová zkratka: J

Menu: Mesh » Vertices » Vertex Connect or CtrlV » Vertex Connect

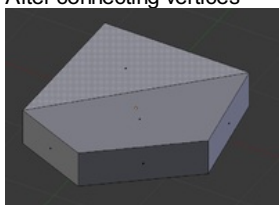
Vertex Connect takes two edges that share a face, and creates an edge between the two, splitting the face into two new faces.



Selected vertices before  
connecting



After connecting vertices



Two faces created from

vertex connect operation



## Vertex Slide

Mód: editační režim

Panel: Editing context → Mesh Tools

Klávesová zkratka: ⇧ ShiftV » Vertex Slide

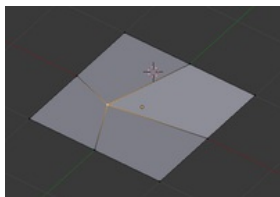
Menu: Mesh » Vertices » Vertex Slide or CtrlV » Vertex Slide

Vertex Slide will transform a vertex along one of its adjacent edges. Use ⇧ ShiftV to enter tool. Highlight the desired edge by moving the mouse, then confirm with LMB . Drag the cursor to specify the position along the line formed by the edge, then LMB  again to move the vertex. There are three options available by holding the following keys:

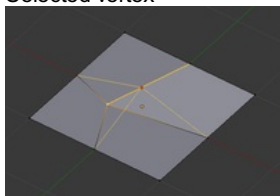
Snap to Midpoint ⇧ Shift

Snap to Endpoint Alt

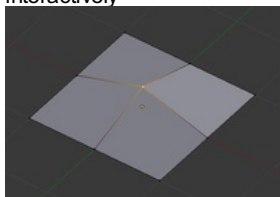
Snap and Merge to Endpoint Control



Selected vertex



Positioning vertex interactively



Repositioned vertex

## Vyhlazení (Smooth)

Mód: editační režim

Panel: Editing context → Mesh Tools

Klávesová zkratka: CtrlV » Smooth vertex

Menu: Mesh » Vertices » Smooth, Specials » Smooth or Vertex Specials » Smooth

This will apply once the [Smooth Tool](#).

## Vytvoření z vrcholu rodiče

Mód: editační režim

Klávesová zkratka: CtrlP

Menu: Mesh » Vertices » Make Vertex Parent

This will parent the other selected object(s) to the vertices/edges/faces selected, as described [here](#).

## Přidání háku (Hook)

Mód: editační režim

Klávesová zkratka: CtrlH

Menu: Mesh » Vertices » Add Hook

Adds a [Hook Modifier](#) (using either a new empty, or the current selected object) linked to the selection. Note that even if it appears in the history menu, this action cannot be undone in Edit mode – probably because it involves other objects...

## Mísení ze tvarů, množení tvarů

Mód: editační režim

Klávesová zkratka: W » AltBlend from shape or CtrlV » Blend From Shape, and W » AltShape propagate or CtrlV » Shape Propagate

Menu: (Vertex) Specials » Blend From Shape and Vertex Specials » Shape Propagate

These are options regarding [shape keys](#).

## Úpravy hran

Modelování a úpravy hran v editačním režimu patří k velice důležitým oblastem modelování. Bez znalosti modelování tohoto typu je zřídka možné docílit kýženého cíle modelu. Blender nabízí mnoho způsobů, jak upravovat hrany objektu. Následující kapitoly popisují základní z nich.

## Vytvoření hrany/stěny

Mód: Edit mode

Klávesová zkratka: F

Menu: Mesh » Edges » Make Edge/Face

Takto je vytvořena hrana, nebo stěna podle výběru předchozích prvků. Tento nástroj je již diskutován v kapitole základů [Základy editace](#).

## Nastavení atributů hran

Hrany mají několik vlastností které jim udílí chování po použití dalších nástrojů.

### Mark Seam and Clear Seam

Mode: Edit mode (Vertex or Edge select modes)

Hotkey: CtrlE1 NumPad and CtrlE2 NumPad

Menu: Mesh » Edges » Mark Seam/Clear Seam (or the same options in Edge Specials menu)

Seams are a way to create separations, "islands", in UV maps. See the [UVTexturing section](#) for more details. These commands set or unset this flag for selected edges.

### Mark Sharp and Clear Sharp

Mode: Edit mode (Vertex or Edge select modes)

Hotkey: CtrlE1 NumPad and CtrlE2 NumPad

Menu: Mesh » Edges » Mark Seam/Clear Seam (or the same options in Edge Specials menu)

The Sharp flag is used by the [EdgeSplit modifier](#), which is part of the smoothing technics. As seams, it is a property of edges, and these commands set or unset it for selected ones.

### Adjust Bevel Weight

Mode: Edit mode (Vertex or Edge select modes)

Hotkey: Ctrl⇧ ShiftE

Menu: Mesh » Edges » Adjust Bevel Weight

This edge property (a value between **0.0** and **1.0**) is used by the [Bevel modifier](#) to control the bevel intensity of the edges. This command enters an interactive mode (a bit like transform tools), where by moving the mouse (or typing a value with the keyboard) you can set the (average) bevel weight of selected edges.

### Crease SubSurf

Mode: Edit mode (Vertex or Edge select modes)

Hotkey: ⇧ ShiftE

Menu: Mesh » Edges » Crease SubSurf

This edge property (a value between **0.0** and **1.0**) is used by the [Subsurf modifier](#) to control the sharpness of the edges in the subdivided mesh. This command enters an interactive mode (a bit like transform tools), where by moving the mouse (or typing a value with the keyboard) you can set the (average) crease value of selected edges.

## Posuv hrany


Mode: Edit mode (Vertex or Edge select modes)

Hotkey: CtrlE » 6 NumPad

Menu: Mesh » Edges » Slide Edge (or the same option in Edge Specials menu)

Slides one or more edges across adjacent faces with a few restrictions involving the selection of edges (i.e. the selection must make sense, see below.)



EvenE

Forces the edge loop to match the shape of the adjacent edge loop. You can flip to the opposite vertex using F. Use Alt Wheel 


to change the control edge.

Flip F

When Even mode is active, this flips between the two adjacent edge loops the active edge loop will match

LMB  confirms the tool, and RMB  or Esc cancels.

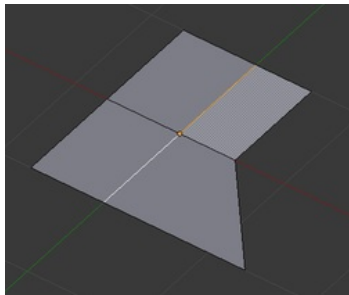
This tool has a factor, which is displayed in the 3D View footer and in the Tool Shelf (after confirmation). A numerical value between -1 and 1 can be entered for precision.

In *Proportional* mode, Wheel  or ← and → changes the selected edge for calculating a proportion. Unlike *Percentage* mode, *Proportional*

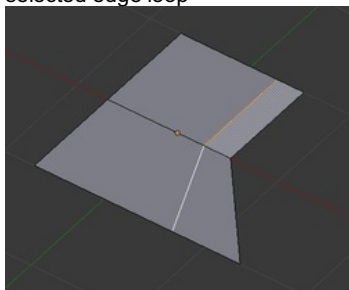
Holding Ctrl or ⇧ Shift control the precision of the sliding. Ctrl snaps movement to 10% steps per move and ⇧ Shift snaps movement to 1% steps. The default is 5% steps per move.

## Použití

By default, the position of vertices on the edge loop move as a percentage of the distance between their original position and the adjacent edge loop, regardless of the edges' lengths.



selected edge loop




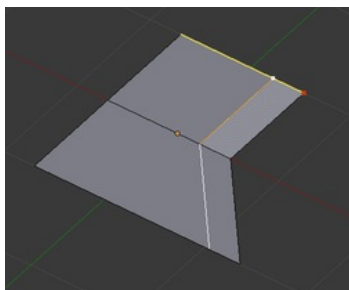
Repositioned edge loop

## Even mode

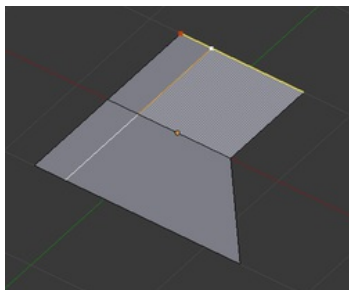
*Even* mode keeps the shape of the selected edge loop the same as one of the edge loops adjacent to it, rather than sliding a percentage along each perpendicular edge.

In *Even* mode, the tool shows the position along the length of the currently selected edge which is marked in yellow, from the vertex that as an enlarged red marker. Movement of the sliding edge loop is restricted to this length. As you move the mouse the length indicator in the header changes showing where along the length of the edge you are.

To change the control edge that determines the position of the edge loop, use the Alt Wheel  to scroll to a different edge.



Even mode enabled



Even mode with flip enabled

Moving the mouse moves the selected edge loop towards or away from the start vertex, but the loop line will only move as far as the length of the currently selected edge, conforming to the shape of one of the bounding edge loops.

### Omezení & Workarounds

There are restrictions on the type of edge selections that can be operated upon. Invalid selections are:

#### Loop crosses itself

This means that the tool could not find any suitable faces that were adjacent to the selected edge(s). (*Loop crosses*) is an example that shows this by selecting two edges that share the same face. A face cannot be adjacent to itself.

#### Multiple edge loops

The selected edges are not in the same edge loop, which means they don't have a common edge. You can minimize this error by always selecting edges end to end or in a "Chain". If you select multiple edges just make sure they are connected. This will decrease the possibility of getting looping errors.

#### Border Edge

When a single edge was selected in a single sided object. An edge loop can not be found because there is only one face. Remember, edge loops are loops that span two or more faces.

A general rule of thumb is that if multiple edges are selected they should be connected end to end such that they form a continuous chain. This is *literally* a general rule because you can still select edges in a chain that are invalid because some of the edges in the chain are in different edge loops.

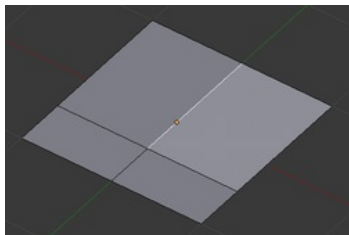
## Rotace hrany

Mode: Edit mode (Vertex or Edge select modes)

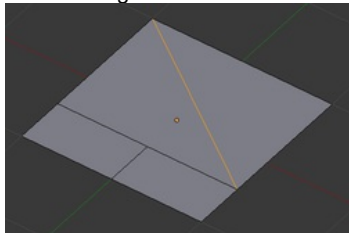
Hotkey: CtrlE » Rotate Edge CW and CtrlE » Rotate Edge CCW

Menu: Mesh » Edges » Rotate Edge CW / Rotate Edge CCW

Rotating an edge clockwise or counter-clockwise spins an edge between two faces around their vertices. This is very useful for restructuring a mesh's topology. The tool can operate on one explicitly selected edge, or on two selected vertices or two selected faces that implicitly share an edge between them.



selected edge



Edge, rotated CW

### Použití výběru stěn

To rotate an edge based on faces you must select two faces, (*Adjacent selected faces*), otherwise Blender notifies you with an error message, "ERROR: Could not find any select edges that can be rotated". Using either Rotate Edge CW or Rotate Edge CCW

will produce exactly the same results as if you had selected the common edge shown in (*Selected edge rotated CW and CCW*).

## Smazání smyčky hran

Mode: Edit mode (Vertex or Edge select modes)

Hotkey: X/Del » 7 NumPad

Menu: Mesh » Edges » Delete Edge Loop (or Edge Loop option of the Erase menu)

Delete Edge Loop allows you to delete a selected edge loop if it is between two other edge loops. This will create one face-loop where two previously existed.

### Note

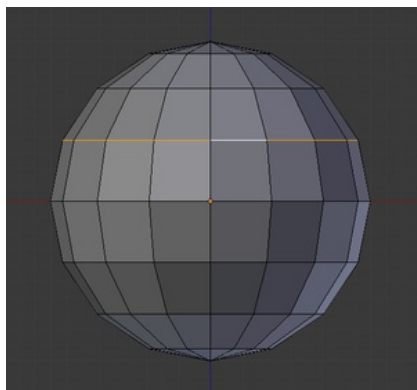
The Edge Loop option is very different to the Edges option, even if you use it on edges that look like an edge loop. Deleting an edge loop merges the surrounding faces together to preserve the surface of the mesh. By deleting a chain of edges, the edges are removed, deleting the surrounding faces as well. This will leave holes in the mesh where the faces once were.

## Limitations & Workarounds

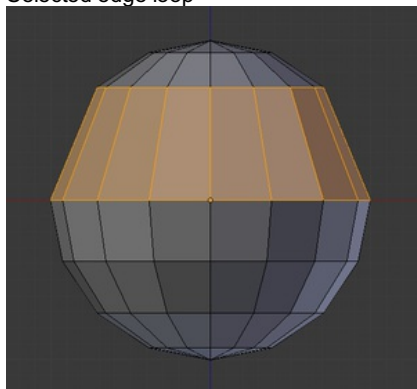
You can only delete **one** (part of) an edge loop at a time, else the tool will show you one of the error messages “loop crosses itself” or “Was not a single edge loop”. In general, the same restrictions as those of Edge Slide apply, see [above](#) for more details.

## Example

The selected edge loop on the UV Sphere has been deleted and the faces have been merged with the surrounding edges. If the edges had been deleted by choosing Edges from the (*Erase Menu*) there would be an empty band of deleted faces all the way around the sphere instead.



Selected edge loop



Edge loop deleted

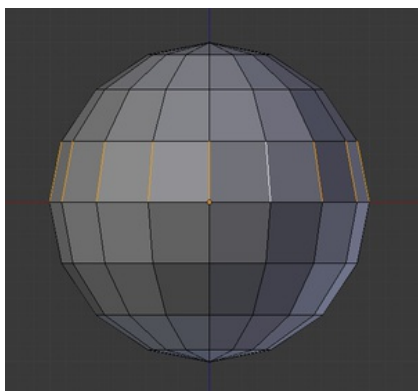
## Collapse

Mode: Edit mode

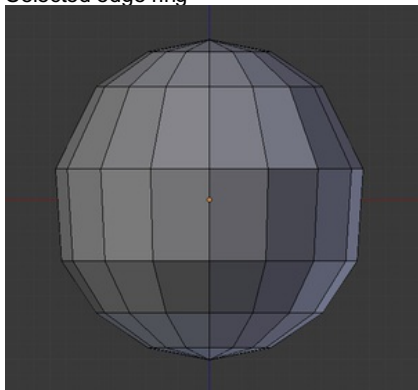
Hotkey: AltM » 3 NumPad

Menu: Mesh » Delete » Edge Collapse

This takes a selection of edges and for each edge, merges its two vertices together. This is useful for taking a ring of edges and collapsing it, removing the face loop it ran through.



Selected edge ring



Edge ring collapsed

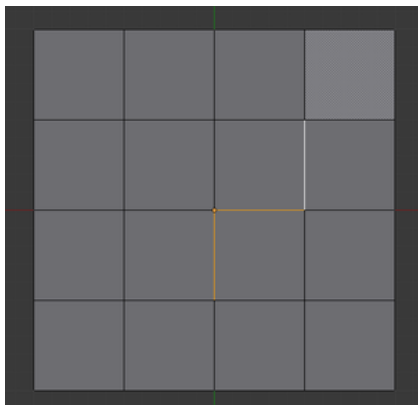
## Edge Split

Mode: Edit mode

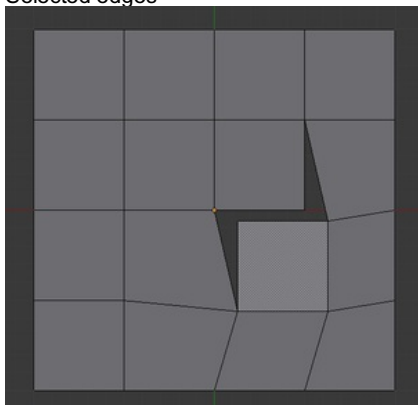
Hotkey: CtrlE » Edge Split

Menu: Mesh » Edges » Edge Split

Edge split is similar to the rip tool. When two or more touching interior edges, or a border edge is selected when using Edge split, a hole will be created, and the selected edges are duplicated to form the border of the hole



Selected edges



Adjacent face moved to reveal hole left by split

Nástroje pro stěny

Následující nástroje jsou určeny pro úpravy a manipulace se stěnami (tj. plochami v šit'ovine objektu, které jej ohraničují).

## Vytváření stěnes

### Vytvoření hrany/stěny

Mode: Edit mode

Hotkey: F

Menu: Mesh » Faces » Make Edge/Face

This will create an edge or some faces, depending on your selection. It is detailed in the [Basic Editing page](#).

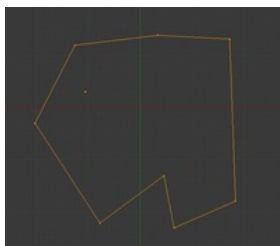
### Vyplnění

Mode: Edit mode

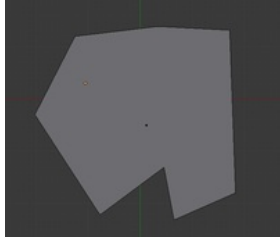
Hotkey: AltF

Menu: Mesh » Faces » Fill/Beautify Fill

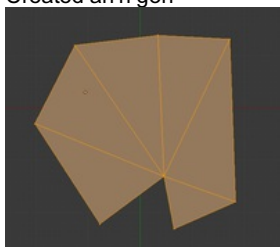
The Fill option will create *triangular* faces from any group of selected edges or vertices, *as long as they form one or more complete perimeters*. These triangles are not always created “in the best way” it is possible (there might results in long and thin triangles) – but I don't know the algorithm used by this tool...



A closed perimeter of edges



Filled using shortcut F.  
Created an n-gon



Filled using fillAltF

### Krásné vyplnění

Mode: Edit mode

Hotkey: Alt⇧ ShiftF

Menu: Mesh » Faces » Fill/Beautify Fill

Beautify Fill works only on selected existing faces. It rearrange selected triangles to obtain more “balanced” ones (i.e. less long thin triangles).



Text converted to a mesh

Result of Beauty Fill,Alt⇧  
ShiftF

## Konverze čtyřúhelníků na trojúhelníky

Mode: Edit mode

Hotkey: CtrlT

Menu: Mesh » Faces » Convert Quads to Triangles or Face Specials » Triangulate

As its name intimates, this tool converts each selected quadrangle into two triangles. Remember that quads are just a set of two triangles.

## Konverze trojúhelníků na čtyřúhelníky

Mode: Edit mode

Panel: Mesh Tools (Editing context)

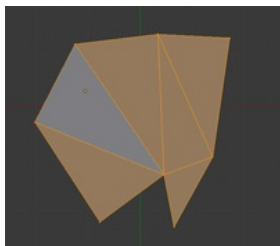
Hotkey: AltJ

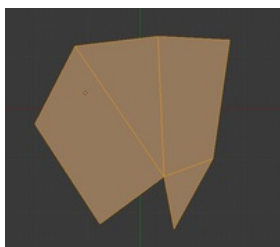
Menu: Mesh » Faces » Convert Triangles to Quads

This tool converts the selected triangles into quads by taking adjacent tris and removes the shared edge to create a quad, based on a threshold. This tool can be performed on a selection of multiple triangles.

This same action can be done on a selection of 2 tris, by selecting them and using the shortcut F, to create a face, or by selecting the shared edge and dissolving it with the shortcut X » Dissolve.

To create a quad, this tool needs at least two adjacent triangles. If you have an even number of selected triangles, it is also possible not to obtain only quads. In fact, this tool tries to create “squarishest” quads as possible from the given triangles, which means some triangles could remain.

Before converting tris to  
quads



After converting tris to quads,  
with a max angle of 30

All the menu entries and hotkey use the settings defined in the Mesh Tools panel:

#### Max Angle

This values (between **0** and **180**) controls the threshold for this tool to work on adjacent triangles. With a threshold of **0.0**, it will only join adjacent triangles that form a perfect rectangle (i.e. right-angled triangles sharing their hypotenuses). Larger values are required for triangles with a shared edge that is small, relative to the size of the other edges of the triangles.

#### Compare UVs

When enabled, it will prevent union of triangles that are not also adjacent in the active UV map. Note that this seems to be the only option working...

#### Compare Vcol

When enabled, it will prevent union of triangles that have no matching vertex color. I'm not sure how this option works – or even if it really works...

#### Compare Sharp

When enabled, it will prevent union of triangles that share a “sharp” edge. I'm not sure either if this option works, and what is the “sharp” criteria – neither the Sharp flag nor the angle between triangles seem to have an influence here...

#### Compare Materials

When enabled, it will prevent union of triangles that do not use the same material index. This option does not seem to work neither...

## Solidify

Mode: Edit mode

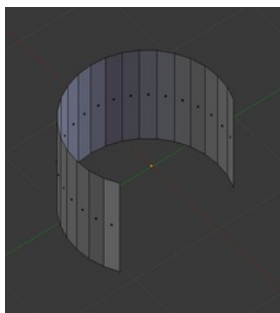
Hotkey: CtrlF » Solidify

Menu: Mesh » Faces » Solidify

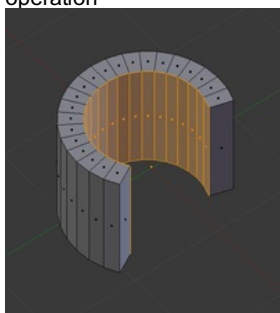
This takes a selection of faces and solidifies them by extruding them uniformly to give volume to a non-manifold surface. This is also available as a [Modifier](#). After using the tool, you can set the offset distance in the Tool Palette.

#### Thickness

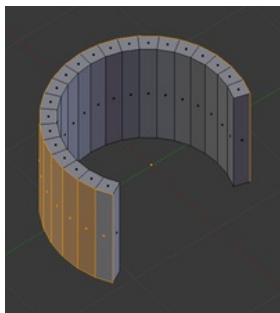
Amount to offset the newly created surface. Positive values offset the surface inward relative to the normals. Negative values offset outward.



Mesh before solidify  
operation



Solidify with a positive  
thickness



Solidify with a negative thickness

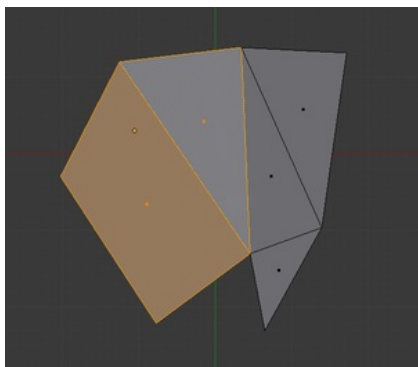
## Rotace hran

Mode: Edit mode

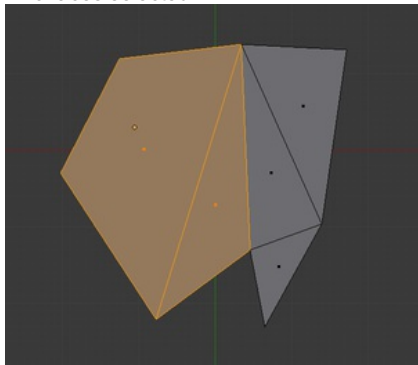
Menu: Mesh » Faces » Rotate Edge CW

This command functions the same edge rotation in edge mode.

It works on the shared edge between two faces and rotates that edge if the edge was selected.



Two faces selected



After rotating edge

See [Rotate Edge CW / Rotate Edge CCW](#) for more information.

## Normály

As normals are mainly a face “sub-product”, we describe there few options here also.

See [Smoothing](#) for additional information on working with face normals.

### Překlopení směru

Mode: Edit mode

Hotkey: W » Flip Normals}

Menu: Mesh » Normals » Flip or Specials » Flip Normals

Well, it will just reverse the normals direction of all selected faces. Note that this allows you to precisely control the direction (**not the orientation**, which is always perpendicular to the face) of your normals, as only selected ones are flipped.

### Přepočítání normál

Mode: Edit mode

Hotkey: CtrlN and ctrl

Menu: Mesh » Normals » Recalculate Outside and Mesh » Normals » RecalculateInside

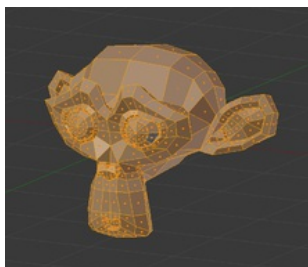
These commands will recalculate the normals of selected faces so that they point outside (respectively inside) the volume that the face belongs to. This volume do not need to be closed. In fact, this means that the face of interest must be adjacent with at least one non-coplanar other face. For example, with a Grid primitive, neither Recalculate Outside nor Recalculate Inside will never modify its normals...

Geometrické deformace

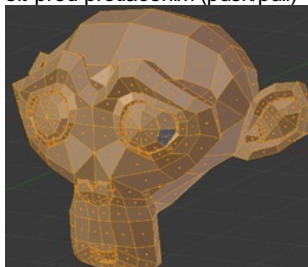
## Vytlačení/vytažení (Push/Pull)

Podobně jako zploštění, zcvrknutí obsahuje tato transformace posuv vybraných elementů podél spojnice mezi nimi a průměrného středu výběru. Všechny translace jsou podle stejné hodnoty a jsou ovládány myší. JE to podobné jako u transformace škálování, ale elementy jsou tímto efektem více deformovány.

Je samozřejmě možné před působením deformace využít kontrolu nad jednotlivými osami ([zámek os x,y,z, Axis Locking](#)), a tak ponechat působení efektu pouze ve směru jedné, nebo dvou os.



síť před protlačení (push/pull)



Vytažená síť pomocí záporného parametru



Stlačená síť pomocí kladného parametru

## Pokřivení (Warp)

Mód: Edit mode

Klávesová zkratka: ⇧ ShiftW

Menu: Mesh/Curve/Surface » Transform » Warp

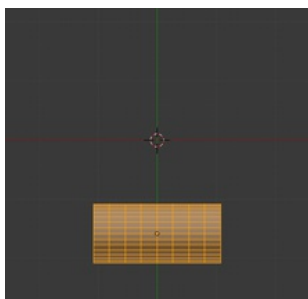
Transformace typu Warp je užitečná ve velice specifických případech. Působí tak, že vybrané prvky pokříví okolo 3D kurzoru. (vždy podle 3D kurzoru, nastavení pívotu nemá vliv). Okamžitý vliv pokřivení je vidět ihned. Body, které jsou vyrovnány svisle s kurzorem zůstanou na svém místě. Vzdálenost každého bodu od 3D kurzoru měřeno **horizontálně** odpovídá poloměru kružnice, po které budou body deformovány. Hodnota parametru pokřivení 360 pokříví a zkroutí objekt natolik, že se podobá kružnici s tím, že původní spojnice vrcholů zůstávají spojnicemi..

Před použitím tohoto deformáčního nástroje umístěte kurzor do míst, kolem kterých se má deformace provést. Aktivujte nástroj, a pak přemístěte kurzor nebo zadejte hodnotu pokřivení..

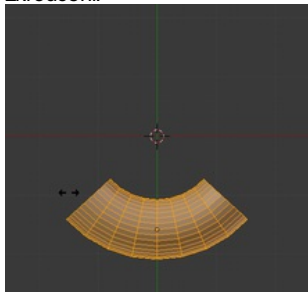
### Příklad

Válec je pokřivený do polokruhového tvaru

- Přepněte pohled na půdorys a přemístěte síť objektu mimo 3D kurzor. Tato vzdálenost definuje poloměr efektu pokřivení.
- Přepněte se do editačního režimu pomocí (⇧ Tab) a pro výběr všech vrcholů stiskněte A. Dále stiskněte ⇧ ShiftW, a tím je aktivován tranfomační nástroj pokřiven-Warp.



Válec před působením zkroucení.



Válec pokřiven malým úhlem.



Válec pokřiven velkým úhlem.



## Smýkání, zkosení (Shear)

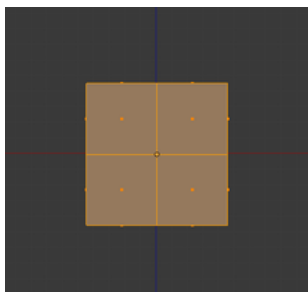
Mode: Edit mode

Hotkey: CtrlAlt⇧ ShiftS

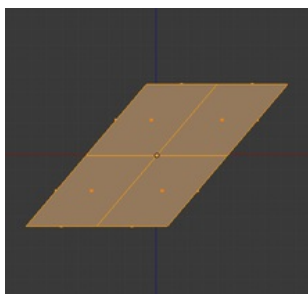
Menu: Object/Mesh/Curve/Surface » Transform » Shear

Transformace typu Shear (dále jen "smyk") používá posuvu elementů jako jsou vrcholy, hrany, řídící body v editačním režimu. Podobně jako ostatní transformační nástroje využívá prostor pohledu a bod pivot. "Smýkání" probíhá podél pohledové osy x procházející bodem pivotu. Všechno, co je "nad" touto osou (tj. má kladnou hodnotu osy y) se bude posouvat ve stejném směru jako kurzor myši (ale vždy souběžně s osou x). A všechno, co je pod touto osou x se bude přesouvat ve směru opačném.

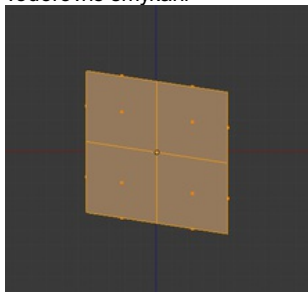
Když se efekt stane aktivní, pohybujte interaktivně myši a tím řídíte zkosení. Pokud požadujete, aby efekt byl účinný i na jiné nežli horizontální úrovni, klikněte MMB  a přesuňte kurzor myši nahoru a dolů. Nebo použijte numerickou hodnotu od 0 do nekonečna. Pro ukončení stiskněte LMB .



před smýkáním



vodorovné smýkání



svislé smýkání

## Do koule (To Sphere)

Mód: Edit modes

Panel: Mesh Tools (Editing context)

Klávesová zkratka: ⇧ ShiftAltS

Menu: Mesh/Curve/Surface » Transform » To Sphere

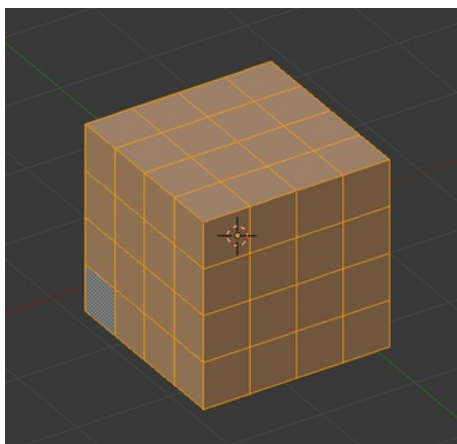
Tento příkaz "zakulatí" vybrané síťové prvky, nebo objekty. Působí pomocí vypočtení průměrné pozice prvků a přesune je k této pozici o průměrnou vzdálenost od středu. Při použití parametru 1 jsou všechny vrcholy umístěny do stejné vzdálenosti od tohoto bodu, dojde k vytvoření kulového tvaru.

Když se stane nástroj aktivní, přesunujte interaktivně myší hodnotu zkreslení, anebo přímo vepište hodnotu ov rozsahu od 0 do 1.

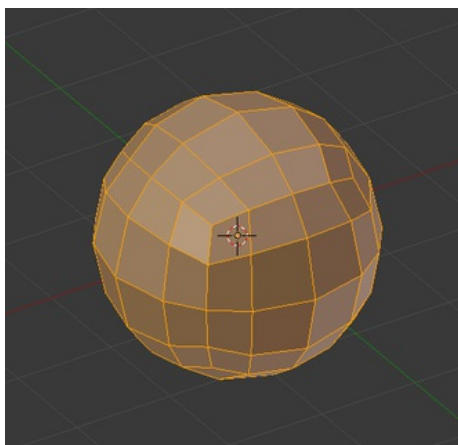
### Příklad

Nejprve vytvořte krychli pomocí [Cube](#).

- Pro přepnutí do editačního režimu stiskněte ⇧ Tab.
- ujistěte se, že jsou vybrané všechny vrcholy sítě krychle - dvojitým stiskem A. Po té přejděte do editování kontextu stiskem F9. Nyní uvidíte panel Mesh Tools.
- Rozdělte (Subdivide) krychli stiskem tlačítka Subdivide v liště nástrojů Mesh Tools, anebo pomocí menu W » Subdivide. Můžete dělit síť kolikrát je potřeba. Samozřejmě při velkém počtu rozdělení bude výsledná síť hladší.
- Nyní stiskněte ⇧ ShiftAltS a přesunujte interaktivně myší hodnotu zkreslení, nebo proporce "sferifikace" (zakulacení), anebo přímo vepište hodnotu jako "1.000" a tím ověřte, že efekt zkreslení bude stejný jako popsáno níže – použitím bodu pivot, tj. středu (mediánu - Median Point).
- Alternativně můžete použít tlačítko "Do koule" (To Sphere) v panelu Mesh Tools. Vyberte "100" pro vytvoření koule. Zde nelze přemístit 3D kurzor – *a tak nezískáte kouli, ale jen část...*



Rozdělení krychle, před efektem



Rozdělena krychle po působení efektu warp

Zrcadlo


Mód: Edit mode

Klávesová zkratka: CtrlM

Menu: Mesh » Mirror » Desired Axis (požadovaná osa)

Nástroj zrcadlo zrcadlí zvolené prvky podél vybrané osy. Nástroj v editační režimu funguje podobně jako zrcadlení objektu [Mirroring in Object mode](#). Funkce je naprosto identická jako škálování s faktorem -1, kdy vrcholy, hrany a stěny jsou zrcadleny podle pívotu a směru vybrané osy s tím, že operace Zrcadlo je rychlejší a šikovnější.

Po té, to se nástroj stane aktivním, je možné vybrat požadovanou osu pro zrcadlení zadáním x,y, nebo z.

Je možné také interaktivně zrcadlit geometrické prvky podržením MMB  za současného tažení požadovaným směrem zrcadlení.

## Osy symetrie (symetrály)

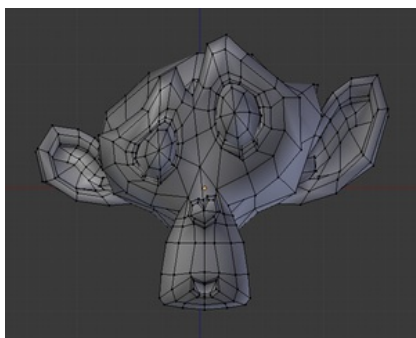
For each transformation orientation, you can chose one of its axis along which the mirroring will occur.

As you can see the possibilities are infinite and the freedom complete: you can position the pivot point at any location around which we want the mirroring to occur, chose one transformation orientation and then one axis on it.

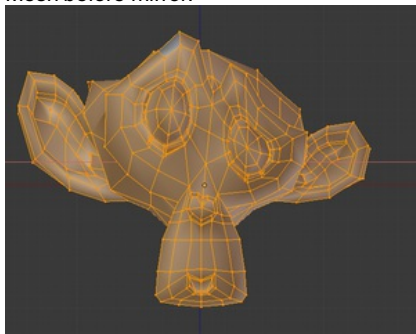
## Pivot point

[Pivot points](#) must be set first. Pivot points will become the center of symmetry. If the widget is turned on it will always show where the pivot point is.

On (*Mirror around the Individual Centers...*) the pivot point default to **median point of the selection of vertices** in Edit mode. This is a special case of the Edit mode as explained on the [pivot point page](#).

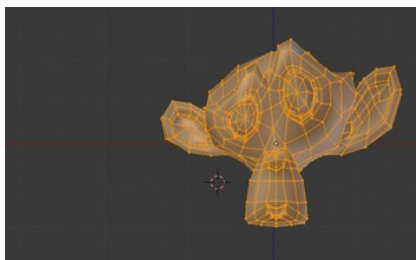


Mesh before mirror.

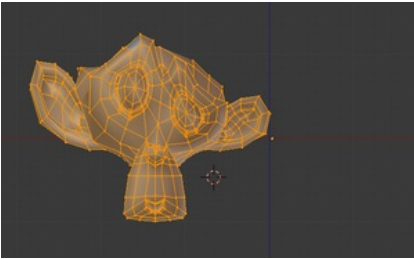


Mesh after mirrored along X axis

On (*Mirror around the 3D Cursor...*) the pivot point is the 3D Cursor, the transformation orientation is Local, a.k.a. the Object space, and the axis of transformation is X.



Mesh before mirror.



Mesh after mirrored along X axis using the 3d cursor as a pivot point

## Transformation orientation

[Transformation Orientations](#) are found on the 3D area header, next to the Widget buttons. They decide of which coordinate system will rule the mirroring.

Hubnutí a tloušťnutí podél normál

Mód: editační režim

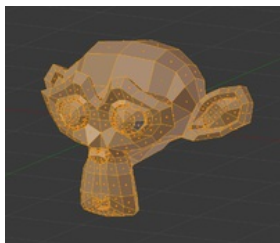
Panel: Mesh Tools (Editing context)

Klávesová zkratka: AltS

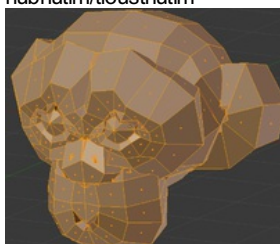
Menu: Mesh » Transform » Shrink/Fatten Along Normals

Tento nástroj umožňuje provádět posouvání vybraných vrcholů, hran a stěn podél jejich normál (kolmice na povrch) a tím je celkově způsobeno prostorové "tloušťnutí", nebo "hubnutí" daného objektu, respektive jeho vybraných částí.

Tento transformační nástroj nebere ohledy na umístění počátku - pivotu, nebo na aktuální orientaci objektu v prostoru.



Původní síťovina před  
hubnutím/tloušťnutím



Nadutá hlava použitím  
kladného čísla



Smrštěná hlava záporným  
číslem

Uhlazení (Smooth)

Mód: editační režim

Panel: Mesh Tools (Editing context, F9)

Klávesová zkratka: CtrlV » Smooth vertex

Menu: Mesh » Vertices » Smooth vertex

Tento nástroj zaoblí a vyhladí vybrané komponenty zprůměrováním úhlů mezi stěnami povrchu. Porch tělesa je po použití funkce hladší, má oblejší všechny hrany /pro představu je možné zkusit mnohonásobné uhlazování krychle. Po několika krocích se postupně z krychle "stává" oblý objekt podobný kouli).

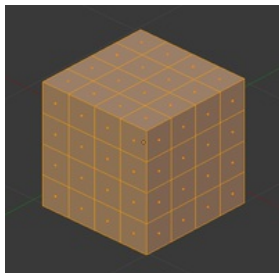
Na použití nástroje je k dispozici volba Tool Shelf:

Number of times to smooth

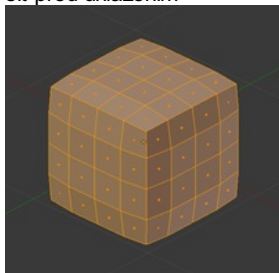
počet iterací (průchodů) vyhlazení

Axes

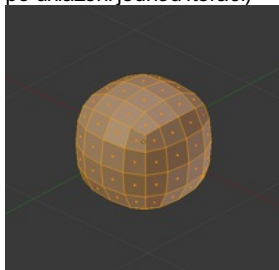
Omezení efektu pouze na některé osy.



síť před uhlazením



po uhlazení jednou iterací)



po uhlazení deseti iteracemi

## Laplaceovo uhlazení

Mód: editační režim

Klávesová zkratka: W » Laplacian Smooth

Details jsou uvedeny v kapitole [Laplaceův uhlazovací modifikátor](#).

Laplaceovo uhlazení je používáno jako alternativa uhlazovacího algoritmu, která dává lepší výsledky s menšími změnami sítě jako celku. Laplaceovo uhlazení je buď tato jednorázová operace nad sítí, nebo také jako nedestruktivní modifikátor.

Poznámka

[Uhlazovací modifikátor](#), který je omezený pouze na skupinu vrcholů (Vertex Group) je nedestruktivní alternativou uhlazovacímu nástroji "smooth tool".

Reálné uhlazení versus Stínové uhlazení

Nezaměňujte tyto dva nástroje. Stínové uhlazení je popsáno na [této stránce](#). Obě funkce nedělají stejnou činnost! Tento nástroj upravuje samotnou síť, redukuje ostrost hran, zatímco nastavení Set Smooth/AutoSmooth je pouze nastavení řízení způsobu stínování povrchu, vytváření *iluze* hladkosti - ale bez modifikace samotné sítě.

## Šum

Mód: Edit mode

Panel: Mesh tools panel (Editing context)

### Poznámka

Noise (šum) je zastaralá funkce. Nedestruktivní alternativou je funkce [Displace Modifier](#), která je více flexibilnější cestou realizace těchto efektů. Hlavní výhodou modifikátoru je, že jeho působení je vratné a může být kdykoli zrušeno, a tak je možné precizně určovat stupeň a směr působení a tohoto posuvného modifikátoru.

Viz také krajina mravence: [add-on](#).

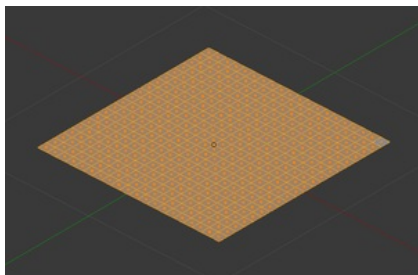
Funkce Šum umožňuje rozprostřít vrcholy sítě pomocí hodnot stupňů šedi v případě prvního slotu textury materiálu, který je připojen ke zpracovávanému objektu.

Síť tak musí mít přiřazený materiál a texturu.

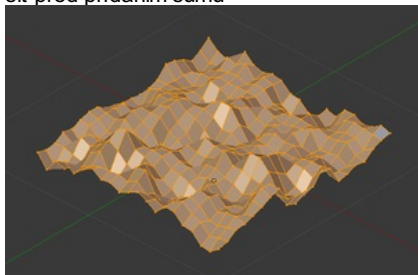
Funkce Šum pouze rozprostírá vrcholy posuvem pouze v ose  $\pm Z$ .

Šum trvale modifikuje síť v souladu s testurou materiálu. Každé kliknutí přidá síť. Pro vytvoření dočasného efektu mapujte texturu na čas mimo renderování. V objektovém/editačním režimu se bude objekt jevit normálně, ale bude renderován deformovaně.

Deformace může být řízena úpravou hodnot v panelu Mapping, anebo přímo v panelu textury. (t.j. Clouds-mraky, Marble-mramor, atp.)



síť před přidáním šumu



síť po přidání šumu s použitím základní textury mraku (cloud)

## Nástroje duplikace sítí

Tato část zahrnuje další nástroje pro duplikaci sítí umožňující přidávat dodatečné geometrické tvary tělesům.

- [Duplikace geometrie](#).
- [Vytažení - Extruze](#).
- [Roztočení - Spin](#).
- [Zkroucení - Screw](#).

## Vícenásobné pohledy

Pokud využíváte jeden z duplikačních nástrojů vz panelu Mesh Tools, Blender nedokáže určit, v jakém z více otevřených pohledů chcete nástroj použít. Pro tyto nástroje je důležitý určený pohled a dokud není aktivován, kurzor je opředený sadou otazníků (to je hezký překlad, že?) - pak pouze stačí kliknout do pohledu, který chcete využít.

Duplikovat

Mode: Edit režim

Hotkey: ⇧ ShiftD

Menu: Mesh » Duplicate

Tento nástroj jednoduše duplikuje vybrané elementy bez vytvoření nějaké vazby, linky s ostatními částmi sítě (narozdíl například od funkce extrude-protažení). Duplikát je umístěn na stejném místě jako originál s tím, že zůstane vybrán pouze duplikát a kurzor je přepnut do režimu přesunu (a tím je možné okamžitě zkopírovaný element přesouvat).

V nástrojové liště Tool Shelf je nastavení pro Vektor posuvu, Proporční editaci, režim duplikace (non-functional?) a omezení os - Axis Constraints.

Duplikované prvky odpovídají stejné skupině vrcholů ([vertex groups](#)) jako jejich originály. Úplně stejně i pro materiálové popisy ([material indices](#)), ostron hran (Sharp) a příznaky Seam a další vlastnosti vrcholů/hran/stěn...

Vytáhnout (Extrude)

## Vytáhnout region

Mód: Edit mode

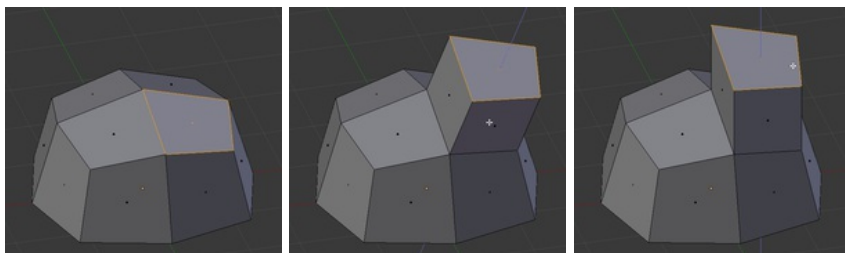
Panel: Mesh Tools » Extrude

Klávesová zkratka: E or AltE

Menu: Mesh » Extrude Region

Jedním z velice užívaných nástrojů při modelování sítě je nástroj Vytažení - Extrude. Umožňuje vytvoření souběžných stěn, hran, válců z kruhu, a podobně. Je jeden z nejvíce využívaných nástrojů v Blenderu vůbec. Použití Extrude je jednoduché, přímočaré a velice efektivní.

Výběr prvků je vytážen podél společné normály vybraných prvků. V každém případě může být směřován směr vytážení (vytahování) podíl konkrétní osy (tj. X omezi na posuv podél X, podobně  $\oplus$  ShiftX na rovinu YZ.. Pokud vytahujeme prvek objektu podél normály a požadujeme na příklad omezení na směr vytážení osou Z, musíme stisknout Z dvakrát. Poprvé pro uvolnění implicitně nastaveného pohybu podél normály a podruhé určení směru Z..



Vybraná stěna

během vytahování

a s nastavením osy Z

Přestože proces vytážení je docela intuitivní, některé principy jsou diskutovány dále:

- Algoritmus určuje venkovní smyčku k vytážení. To je jaký počet vybraných hran bude vytážen a změněn do stěn. Implicitně (viz níže) algoritmus určí, že hrany odpovídající dvěma, nebo více stěnám je interní a tak není součástí smyčky..
- Hrany ve smyčce hran nse pozmění ve stěnu..
- If the edges in the edge-loop belong to only one face in the complete mesh, then all of the selected faces are duplicated and linked to the newly created faces. For example, rectangles will result in parallelepipeds during this stage.
- In other cases, the selected faces are linked to the newly created faces but not duplicated. This prevents undesired faces from being retained "inside" the resulting mesh. This distinction is extremely important since it ensures the construction of consistently coherent, closed volumes at all times when using Extrude.
- When extruding completely closed volumes (like e.g. a cube with all its six faces), extrusion results merely in a duplication, as the volume is duplicated, without any link to the original one.
- Edges not belonging to selected faces, which form an "open" edge-loop, are duplicated and a new face is created between the new edge and the original one.
- Single selected vertices which do not belong to selected edges are duplicated and a new edge is created between the two.

## Individuální vytážení

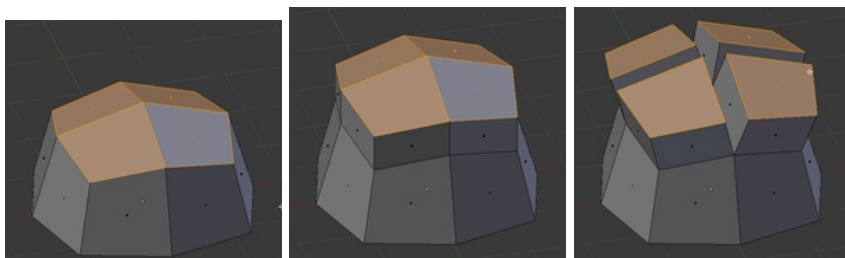
Mode: Edit mode

Panel: Mesh Tools » Extrude Individual

Hotkey: AltE

Menu: Mesh » Extrude Individual

Extrude Individual allows you to extrude a selection of multiple faces as individuals, instead of as a region. The faces are extruded along their own normals, rather than their average. This has several consequences: first, "internal" edges (i.e. edges between two selected faces) are no longer deleted (the original faces are).



Selection of multiple faces

Extruded using extrude region

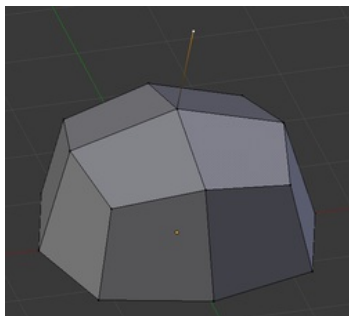
Extruded using Extrude Individual

## Vytažení pouze hran a vrcholů

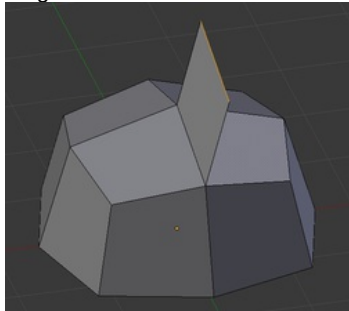
Mode: Edit mode, Vertex and Edge

Hotkey: AltE

If vertices are selected while doing an extrude, but they do not form an edge or face, they will extrude as expected, forming a non-manifold edge. Similarly, if edges are selected that do not form a face, they will extrude to form a face.



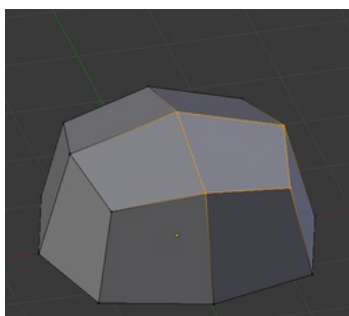
Single vertex extruded



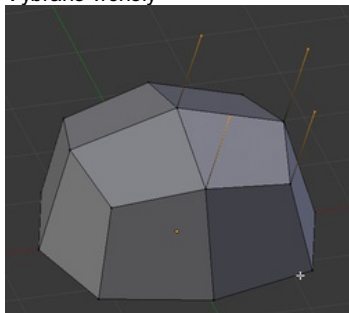
Single edge extruded

When a selection of vertices forms an edge or face, it will extrude as if the edge was selected. Likewise for edges that form a face.

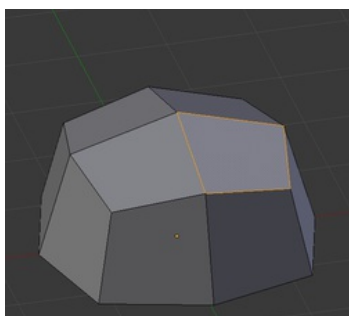
To force a vertex or edge selection to extrude as a vertex or edge, respectively, use AltE to access the Extrude Edges Only and Vertices Only.



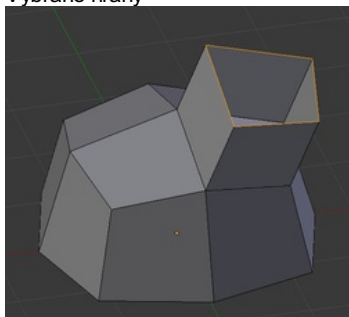
Vybrané vrcholy



Vytažení pouze vrcholů



Vybrané hrany



Vytažení pouze hran

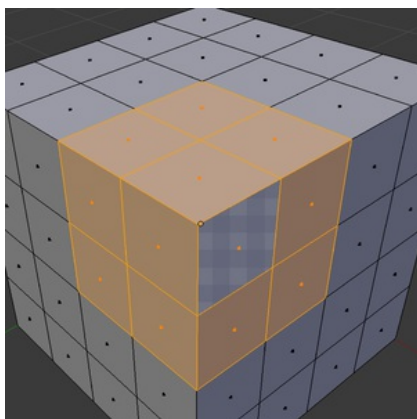
Vsazení (Inset)

Mód: Edit mode

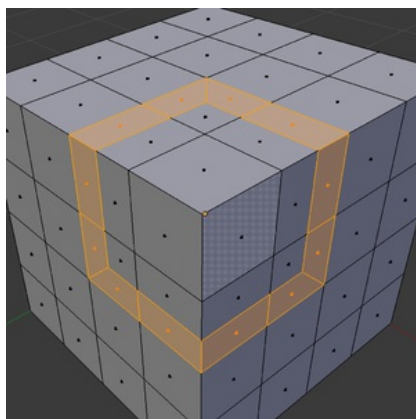
Klávesová zkratka: I

Menu: Mesh » Faces » Inset nebo CtrlF » Inset

Tento nástroj používá právě vybraných stěn a vytvoří další stěny vsazené do vybraných s nastavitelnou tloušťkou a hloubkou. Hloubku vsazení je možné určit pomocí modálního okna podržením Ctrl.

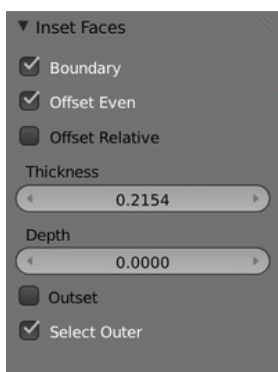


Výběr stěn pro vsazení



Výběr stěn se vsazením

## Volby



Nastavení operace Inset

### Boundary

Okraj. Určuje, zdali bude mít vsazení otevřené hrany, či nikoli.

### Offset Even

Vyrovnaný. Měřítka posuvu určující tloušťku.

### Offset Relative

Měřítka posuvu podle obklopující geometrie.

### Thickness

Nastavení tloušťky vsazení.

### Depth

Zdvížení nebo snížení nového vsazení pro přidání "hloubky".

### Outset

Vytvoří outset spíše nežli vsazení.

### Select Outer

Přepíná, která strana vsazení bude po operaci aktivní.

Spin (roztočení)

Mód: Edit mode

Panel: Mesh Tools (Editing kontext)

Použitím nástroje Spin je možné vytvořit skupinu objektů, které budou vytvářet on a lathe (z tohoto důvodu je někdy proces popisován slovy "lathe"-tool nebo "sweep"-tool). V podstatě jde o druh cirkulárního protažení vybraných prvků s centrováním v kurzoru 3D a torace kolem osy kolmé na rovinu pohledu...

- Bod pohledu určí osu, kolem které bude roztočení uskutečněno...
- Centrem rotace je pozice kurzoru 3D.

Zde je jeho nastavení:

Steps (kroky)

Určuje kolika kopiemi dokola bude uskutečněno roztočení ("sweep").

Dupli

Pokud je zapnuto, zůstanou po rotaci původně vybrané prvky oddělenou skupinou v síti (tj. nepřipojené k výsledku operace).

Angle (úhel)

surčuje úhel roztočení ve stupních (t.j. nastavení **180 °** pro polovinu otočky).

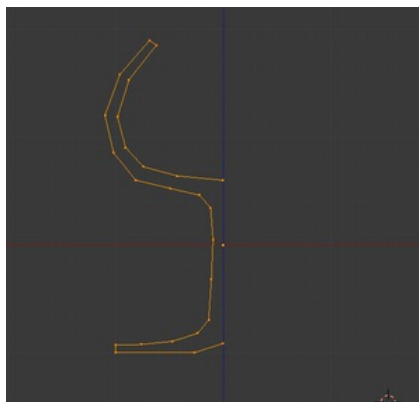
Center

Určuje střed roztočení. Implicitně využije pozici kurzoru.

Axis (osy)

Určuje osy roztočení. Implicitně využije osy pohledu.

## Příklady

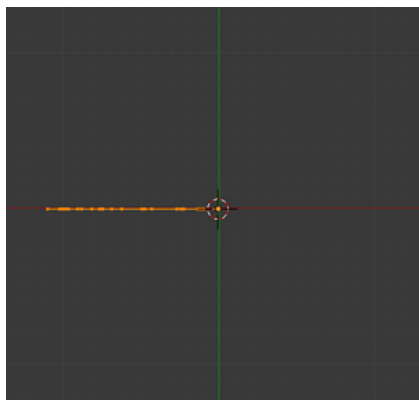


Profil sklenice.

First, create a mesh representing the profile of your object. If you are modeling a hollow object, it is a good idea to thicken the outline. (*Glass profile*) shows the profile for a wine glass we will model as a demonstration.

In Edit mode, with all the vertices selected, access the Editing context (F9). Set the Degr numeric field to a full 360° sweep.

We will be rotating the object around the cursor in the top view, so switch to the top view with 7 NumPad.

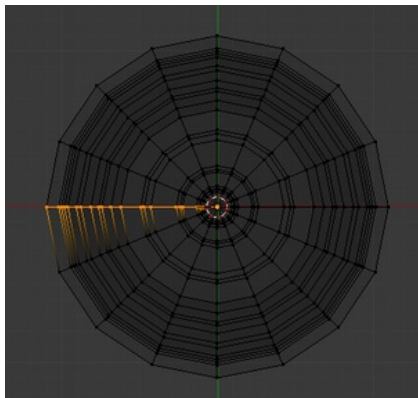


Glass profile, top view in Edit mode, just before spinning.

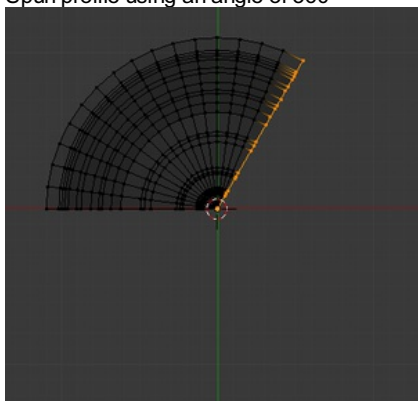
Place the cursor along the center of the profile by selecting one of the vertices along the center, and snapping the 3D cursor to that location with  $\text{⇧ ShiftS}$  » Cursor -> Selection. (*Glass profile, top view in Edit mode, just before spinning*) shows the wine glass profile from top view, with the cursor correctly positioned.

Click the Spin button. If you have more than one 3D view open, the cursor will change to an arrow with a question mark and you will have to click in the window containing the top view before continuing. If you have only one 3D view open, the spin will happen immediately. (*Spun profile*) shows the result of a successful spin.

## Úhel

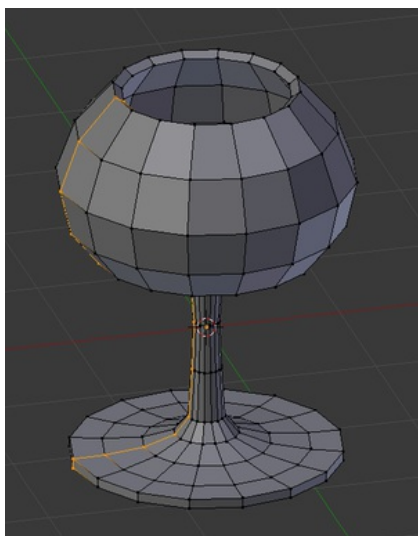


Spun profile using an angle of 360

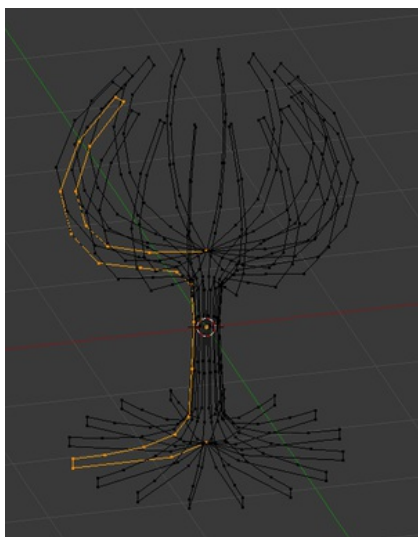


Spun profile using an angle of 120

## Dupli

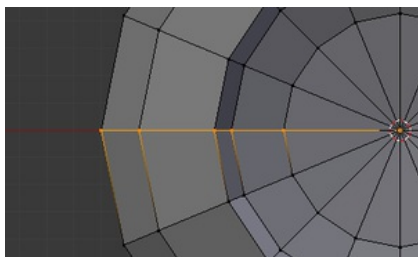


Result of spin operation



Result of Dupli enabled

## Spojit duplikáty



Zdvojené vrcholy

Operace roztočení ponechá vrcholy podél profilu. Je možné vybrat všechny vrcholy pomocí zkratky (B) zobrazené v (*Seam vertex selection*) a použít operaci odstranění duplicit - Remove Doubles.

Notice the selected vertex count before and after the Remove Doubles operation (*Vertex count after removing doubles*). If all goes well, the final vertex count (38 in this example) should match the number of the original profile noted in (*Mesh data – Vertex and face numbers*). If not, some vertices were missed and you will need to weld them manually. Or, worse, too many vertices will have been merged.

### Merging two vertices in one

To merge (weld) two vertices together, select both of them by  $\diamond$  Shift RMB clicking on them. Press S to start scaling and hold down Ctrl while scaling to scale the points down to 0 units in the X, Y and Z axis. LMB to complete the scaling operation and click the Remove Doubles button in the Buttons window, Editing context (also available with W » Remove Doubles). Alternatively, you can use W » Merge from the same Specials menu (or AltM). Then, in the new pop-up menu, choose whether the merged vertex will be at the center of the selected vertices or at the 3D cursor. The first choice is better in our case!

## Přepočítání normál

All that remains now is to recalculate the normals to the outside by selecting all vertices, pressing CtrlN and validating Recalc Normals Outside in the pop-up menu.

poznámka k výrazu "skroutit"

Zkroutit nebo skroutit?

Publikováno: 11. 4. 2007 v 13.26 Zařazen do: Čeština pod lupou Autor: Dalibor Behún

Která z těchto variant je správně? Správné jsou obě podoby, ale nelze je zaměňovat. Mají odlišný význam.

Slovo zkroutit znamená „kroucením dostat do nepřírozené podoby nebo polohy“. Mohu tedy například někomu zkroutit ruku za zády.

Naproti tomu skroutit značí vědomé skroucení za nějakým účelem (podle slovníku „kroucením vytvořit“) – například můžeme z několika vláken skroutit provaz.

--[Quark66](#) 19:14, 1 January 2014 (CET)(Sign)

## Nástroje pro rozdělování sítě

Rozdělování přidává celkovou jemnost povrchu rozřezáváním existujících povrchů a hran na menší části. Systém Blender disponuje pro tuto činnost několika nástroji, jako jsou:

### [Rozdělení \(Subdivide\)](#)

Dělení stěn a hran na menší celky.

### [Rozdělení smyčkou](#)

Vloží smyčku z nových hran mezi existující.

### [Spojení vrcholů](#)

Propojí vybrané vrcholy s hranami, které rozdělují povrchy.

### [Rozdělení nožem](#)

Interaktivním způsobem řeže hrany a stěny.

### [Zkosení \(Bevel\)](#)

Odděluje hrany nebo vrcholy do rozpoložení ploch nebo zaoblení.

Překlad slova "Bevel" do jazyka českého

do češtiny je mnohoznačný, ale přesto vystihuje podstatu tohoto nástroje. Jde o "obroušení" hrany s tím, že jako v praktické situaci při broušení tělesa vzniká nový povrch.

(ze slovníku En-Cz: bevel - kosoúhlý, šikmý, faseta (tech.), pokosník (tech.), sklon (tech.), sražená hrana (tech.), úkos (tech.), zešíkmit, zešíkmený (přid.jm.), zešíkmení, šikmý, kosý, sražený, zkosit, srazit hranu, fasetovat, kuželový, seříznout, šikmá plocha, šikmo uříznout, zkosení.

comment here: [Quark66](#) 05:17, 27 July 2013 (CEST)

Oddělení

Mód: Edit mode

Panel: Mesh Tools (Editing context)

Klávesová zkratka: W » 1 NumPad/2 NumPad

Menu: Mesh » Edges » Subdivide, Specials » Subdivide/Subdivide Smooth

Oddělení rozděljuje vybrané hrany a stěny jejich rozpůlením a přidáním nezbytných dalších vrcholů a hran. Přitom systém dodržuje několik pravidel, jako:

- Pokud je vybrána jedna hrana (Tri mode), trojúhelníky jsou rozděleny na trojúhelníky, čtverce do tří trojúhelníků.
- /Pokud jsou vybrány dvě hrany:
  - a pokud je stěna trojúhelník, je nová hrana vytvořena mezi dvěma novými vrcholy. Trojúhelník je rozdělen na trojúhelník a čtyřúhelník.
  - a pokud je stěna čtyřúhelník a hrany jsou sousedící, dostáváme **tři** možné chování závislosti na nastavení typu řezu rohu (Corner Cut Type) (v menu drop-down za tlačítkem Subdivide v panelu nástrojů sítě (Mesh Tools). Detaily viz níže.
  - a pokud je stěna čtyřúhelník a hrany jsou protilehlé, čtyřúhelník je pouze rozdělen ve dva hranou spojující nové vrcholy.
- Pokud jsou vybrány tři hrany jedné stěny:
  - a stěna je trojúhelník, znamená to, že je vybrána celá stěna, která je rozdělena na čtyři menší trojúhelníky.
  - Pokud je stěna čtyřúhelník, jsou nejprve rozdělny protilehlé hrany. Pak je dělena "střední" hrana podle toho, jaké "podčtyřúhelníky" vzniknou.
- Pokud jsou vybrány čtyři hrany jedné stěny, je plocha rozdělena do čtyř menších čtyřúhelníků.

## Volby

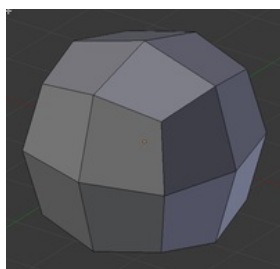
Volby jsou dostupné v panelu nástrojů Tool Panel těsně po spuštění nástroje;

Počet řezů (Number of Cuts)

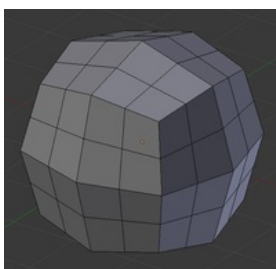
Specifikuje počet skutečných řezů na jednu hrany. Přednastavená je hodnota 1, tedy dělení na dvě části. Hodnota 2 znamená dělení na tři části atd.

Vyhlazení (Smoothness)

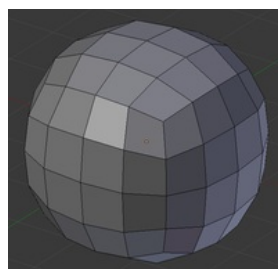
Displaces subdivisions to maintain approximate curvature, The effect is similar to the way the subdivision modifier might deform the mesh.



Síť před rozdělením



Rozdělení bez vyhlazování



a s vyhlazováním = 1.000

Quad/Tri Mode

Poskytuje rozdělení na trojúhelníky namísto polygonů. (viz příklady níže).

Typ řezu rohem

This drop-down menu controls the way quads with only two adjacent selected edges are subdivided

Fan

the quad is sub-divided in a fan of four triangles, the common vertex being the one opposite to the selected edges.

Innervert

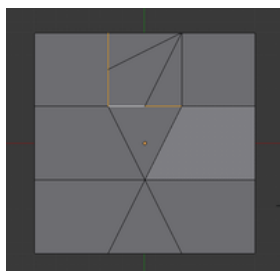
(i.e. "inner vertex"), The selected edges are sub-divided, then an edge is created between the two new vertices, creating a small triangle. This edge is also sub-divided, and the "inner vertex" thus created is linked by another edge to the one opposite to the original selected edges. All this results in a quad sub-divided in a triangle and two quad.

Path

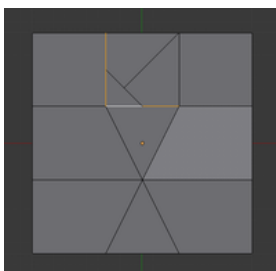
First an edge is created between the two opposite ends of the selected edges, dividing the quad in two triangles. Then, the same goes for the involved triangle as described above.

Straight Cut

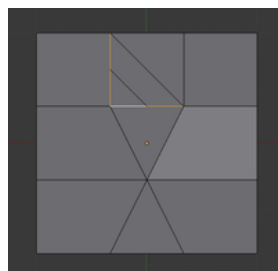
Currently non functioning...



Fan cut type



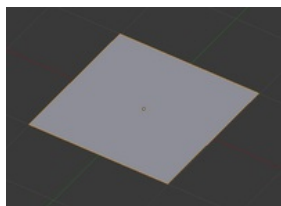
Innervert cut type



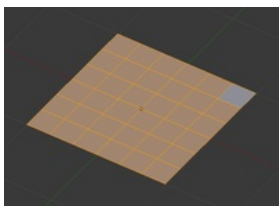
Path cut type

## Fractal

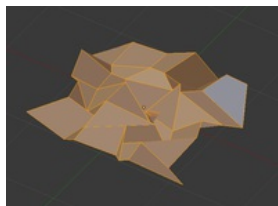
Displaces the vertices in random directions after the mesh is subdivided



Plane before subdivision



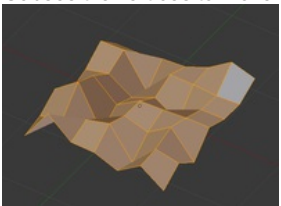
Regular subdivision



Same mesh with fractal added

## Along Normal

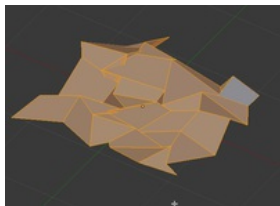
Causes the vertices to move along the their normals, instead of random directions



Along normal set to 1

## Random Seed

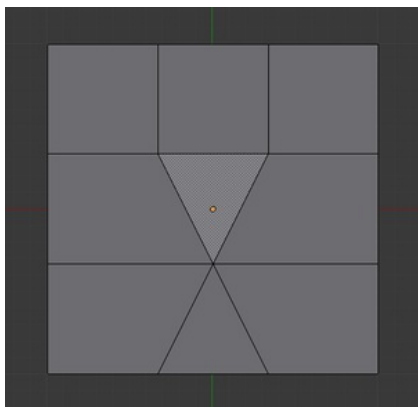
Changes the random seed of the noise function, producing a different result for each seed value.



Same mesh with a different seed value

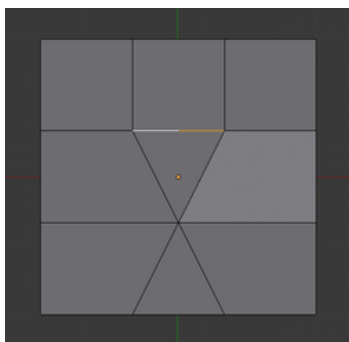
## Příklady

Dále následují různé příklady ilustrující různé možnosti procesů Subdivide a Subdivide Multi s ponechaným výběrem po rozdělení.

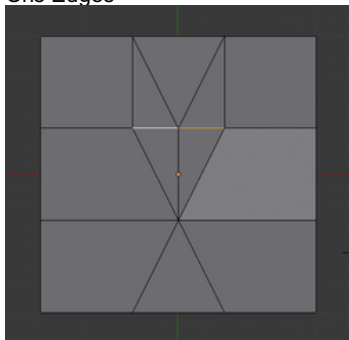


The sample mesh.

## Jedna hrana

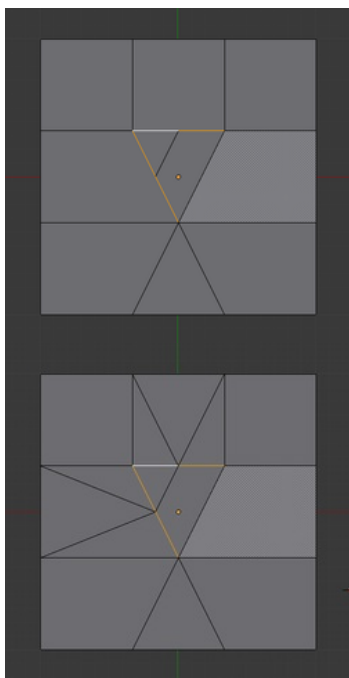


One Edges



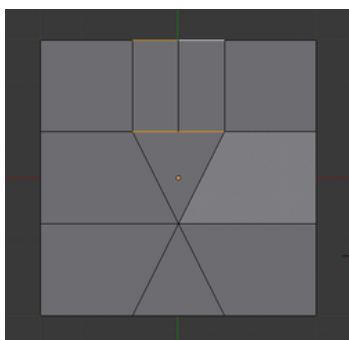
Quad/Tri Mode

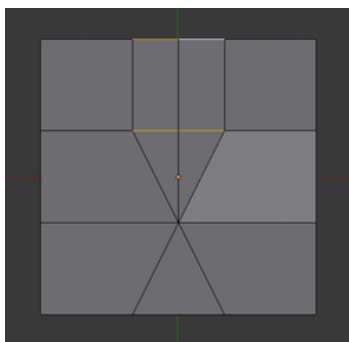
### Dva trojúhelníky



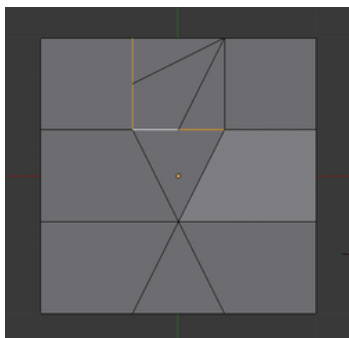
Quad/Tri Mode

### Dvě protilehlé hrany čtyřúhelníku

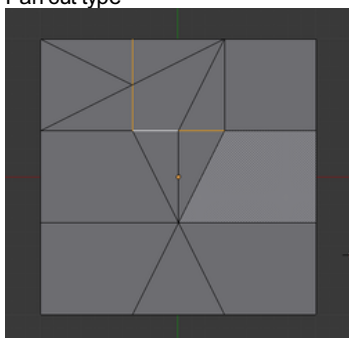




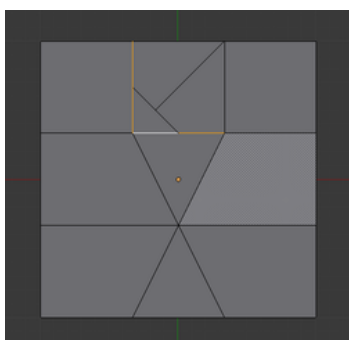
Quad/Tri Mode

**Dvě sousedící hrany čtyřúhelníku**

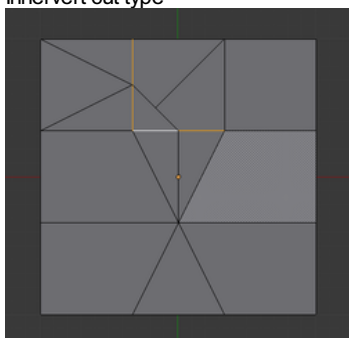
Fan cut type



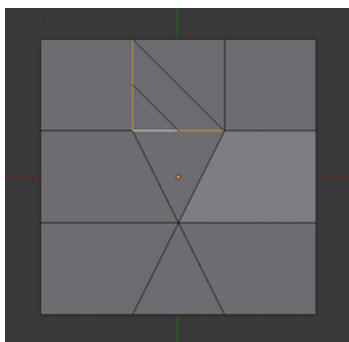
Quad/Tri Mode



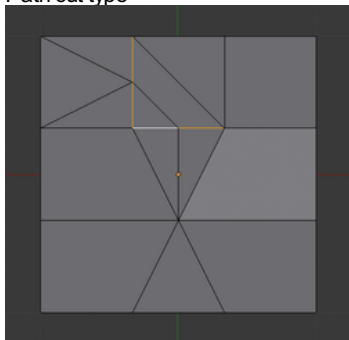
Innervert cut type



Quad/Tri Mode

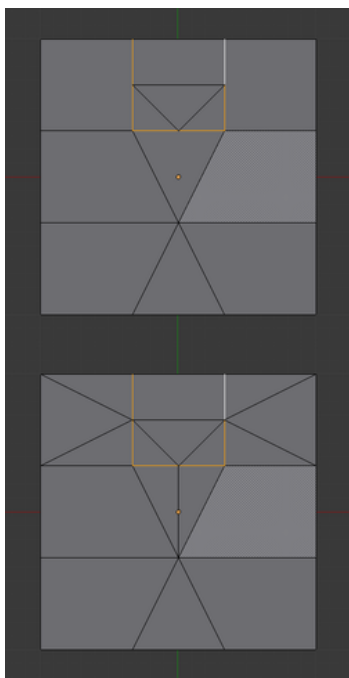


Path cut type



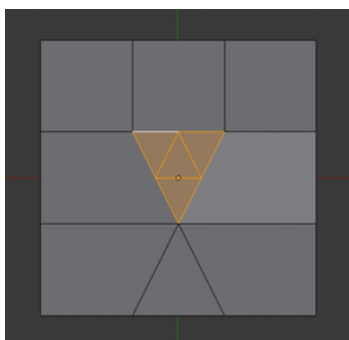
Quad/Tri Mode

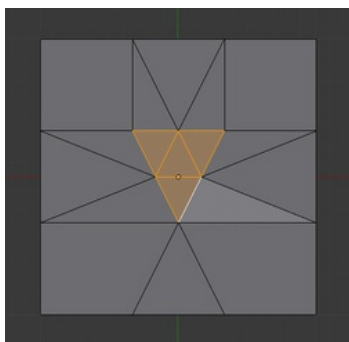
### Tři hrany



Quad/Tri Mode

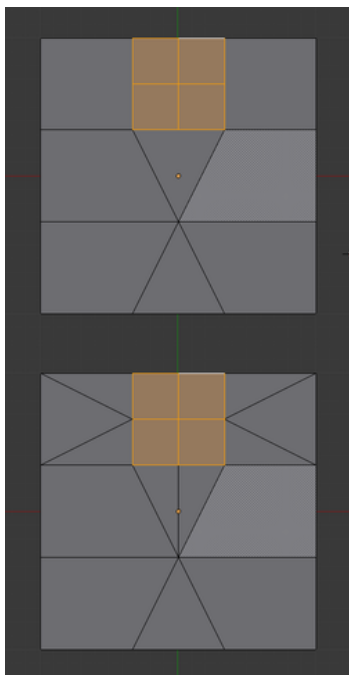
### Režim "Tri mode"





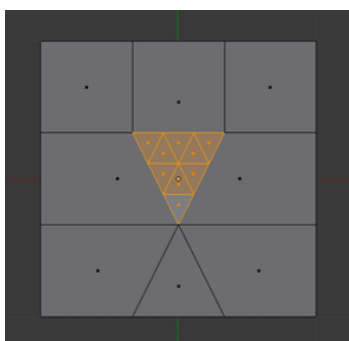
Quad/Tri Mode

### Čtýřúhelník/čtyři hrany

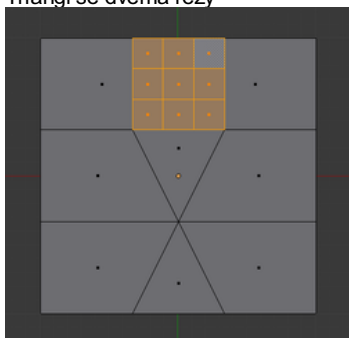


Quad/Tri Mode

### Vícenásobné dělení



Triangl se dvěma řezy



Čtýřúhelník se dvěma řezy



Rozdělení smyčkou

Mód: Edit mode

Panel: Editing context → Mesh Tools

Klávesová zkratka: CtrlR

in progress 10% překládám...




Rozdělení smyčkou (Loop Cut) rozdělí povrchy tělesa vložení nové smyčky, jejíž jednotlivé části jako hrany rozdělí obepípané stěny. Nástroj je interaktivní a obsahuje dva kroky:

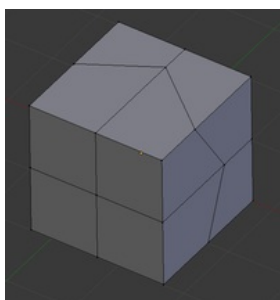
## Využití

### Předzobrazení stříhu

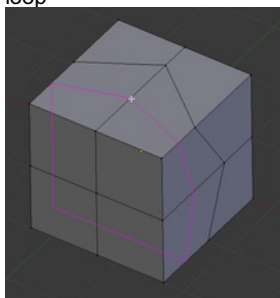
After the tool is activated, move the cursor over a desired edge. The cut to be made is marked with a magenta colored line as you move the mouse over the various edges. The to-be-created edge loop stops at the poles (tris and ngons) where the existing face loop terminates.

### Posun nově vybrané smyčky

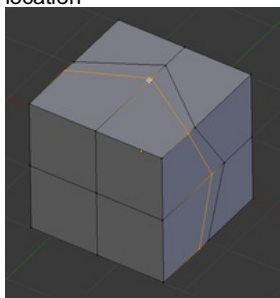
Once an edge is chosen via LMB , you can move the mouse along the edge to determine where the new edge loop will be placed. This is identical to the [Edge Slide tool](#). Clicking LMB  again confirms and makes the cut at the pre-visualized location, or clicking RMB  forces the cut to exactly 50%. This step is skipped when using multiple edge loops (see below)



mesh before inserting edge loop



Preview of edge loop location



Interactive placement of edge loop between adjacent loops

## Volby

Volby jsou dostupné pokud je nástroj právě používán a jsou zobrazeny v hlavičce pohledu 3D


Even E

Only available for single edge loops. This matches the shape of the edge loop to one of the adjacent edge loops. (Detail viz [Edge Slide tool](#))

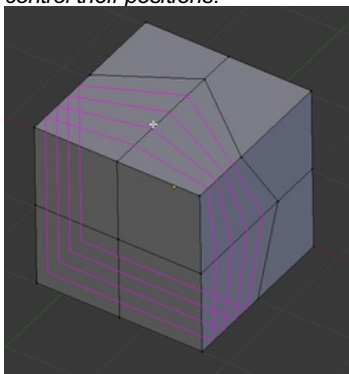
Flip F

When Even is enabled, this flips the target edge loop to match. (See [Edge Slide tool](#) for details)

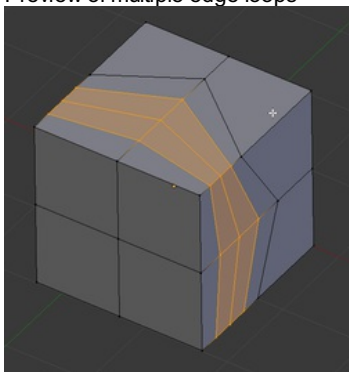
Number of Cuts Wheel  or + NumPad/- NumPad

After activating the tool, but before confirming initial loop location, you can increase and decrease the number of cuts to create, by entering a number with the keyboard, scrolling Wheel  or using + NumPad and - NumPad.


Note that when creating multiple loops, these cuts are uniformly distributed in the original face loop, and *you will not be able to control their positions*.



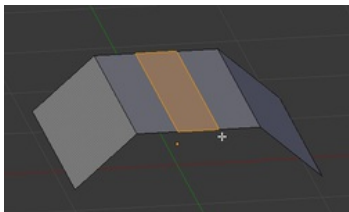
Preview of multiple edge loops



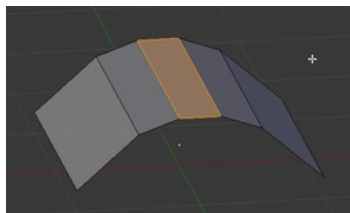
Result of using multiple cuts

Smoothing Alt Wheel 

Smoothing causes edge loops to be placed in an interpolated position, relative to the face it is added to, causing them to be shifted outwards or inwards by a given percentage, similar to the Subdivide Smooth command. When not using smoothing, new vertices for the new edge loop are placed exactly on the pre-existing edges. This keeps subdivided faces flat, but can distort geometry, particularly when using [Subdivision Surfaces](#). Smoothing can help maintain the curvature of a surface once it is subdivided.



Přidání smyčky hran bez vyhlazení



Stejně hranové smyčky ale s hodnotou vyhlazení


## Knife Tool

Mode: Edit mode

Panel: Mesh Tools (Editing context, F9)

Hotkey: K or ⇧ ShiftK

The Knife Tool has been improved for Blender 2.6. It subdivides edges and faces intersected by a user-drawn “knife” line. The tool is now fully interactive, and snaps to edges, cut lines, and vertices, and can create multiple cuts on an edge.

For example, if you wish to cut a hole in the front of a sphere, you select only the front edges, and then draw a line over the selected edges with the mouse. The tool is interactive, and works on primary edges, selected either implicitly by selecting all, or explicitly by box-selecting or ⇧ Shift RMB -clicking a few edges.


Use ⇧ ShiftK or the Select tool in the tool panel to force the knife tool to work only on a selection and in cut-through mode (see below).

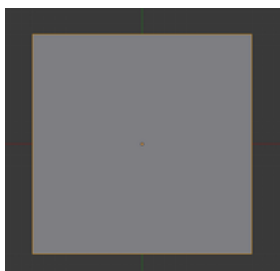
## Usage

When you press K (or ⇧ ShiftK), the Knife tool becomes active.

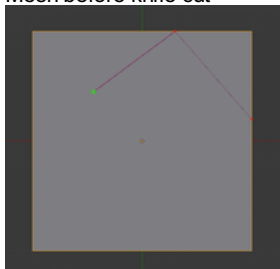
Drawing the cut line

When using Knife Subdivide, the cursor changes to an icon of a scalpel and the header changes to display options for the tool.

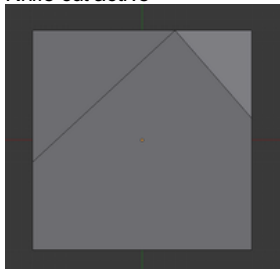
You can draw connected straight lines by clicking LMB .



Mesh before knife cut



Knife cut active

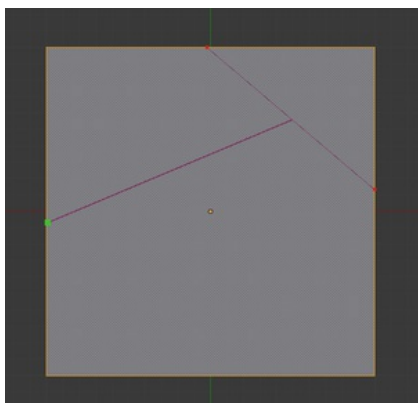


After confirming knife cut

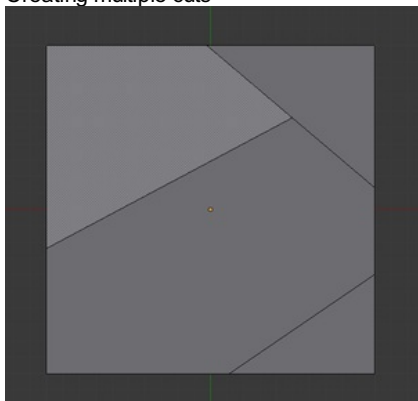
## Options

### New cutE

Begins a new cut. This allows you to define multiple distinct cut lines. If multiple cuts have been defined, they are recognized as new snapping points.



Creating multiple cuts



Result of starting new cuts while in the tool

**Midpoint snap** Ctrl

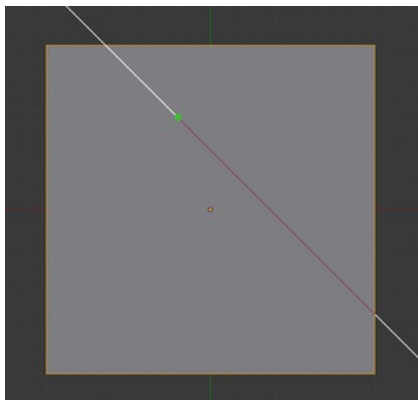
Hold to snap the cursor to the midpoint of edges

**Ignore snap** ⇧ Shift

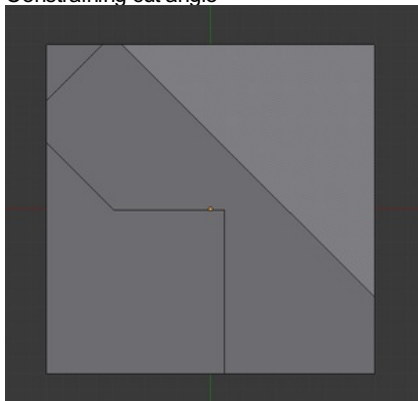
Hold to make the tool ignore snapping.

**Angle constrain** C

Hold to constrain the cut vector to the view in 45 degree increments.



Constraining cut angle




Result of constraining cut angle

**Cut through** Z

Allow the cut tool to cut through to obscured faces, instead of only the visible ones.

## Confirming and selection

Pressing Esc or RMB  at any time cancels the tool, and pressing ↵ Enter confirms the cut, with the following options:

↵ Enter will leave selected every edge except the new edges created from the cut.

## Limitations

If you try to make cuts that end off in the middle of a face, those cuts are ignored. This is a limitation of the current geometry that can be modeled in Blender.

Closed cycles can be cut in the middle of a face, forming holes, but those holes will be connected to the surrounding geometry by two edges, for similar modeling limitation reasons.

In 'cut through' mode, only cut lines that completely cross faces will make cuts.

## Optimizations

With a large mesh, it will be quicker to select a smaller number of vertices—those defining only the edges you plan to split—so that the Knife will save time in testing selected vertices for knife trail crossings.

## Knife Project

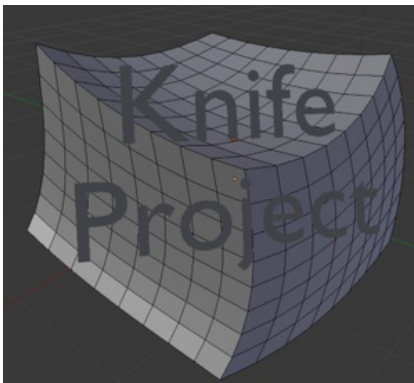
Knife projection is a non-interactive tool where you can use objects to cookie-cut into the mesh rather than hand drawing the line.

This works by using the outlines of other selected objects in edit-mode to cut into the mesh, resulting geometry inside the cutters outline will be selected.

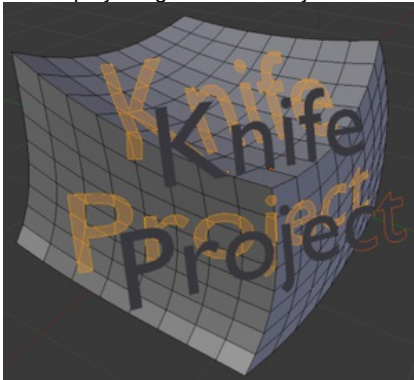
Outlines can be wire or boundary edges.

To use Knife Project, in 'object' mode select the "cutting object" first then shift select the "object to be cut". Now tab into edit mode and press "knife project".

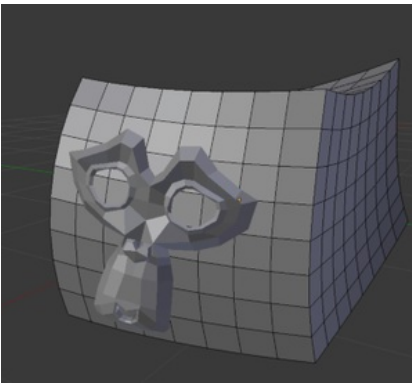
## Examples



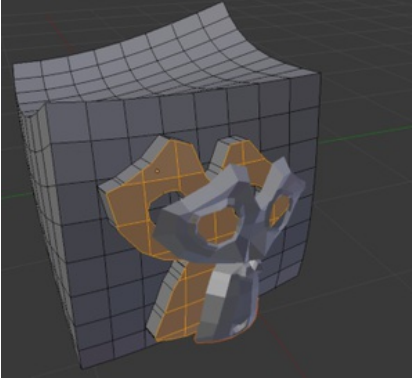
Before projecting from a text object



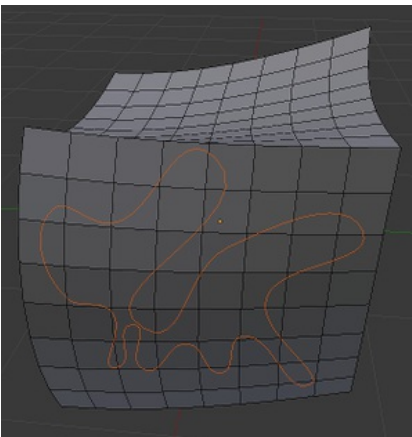
Resulting knife projection



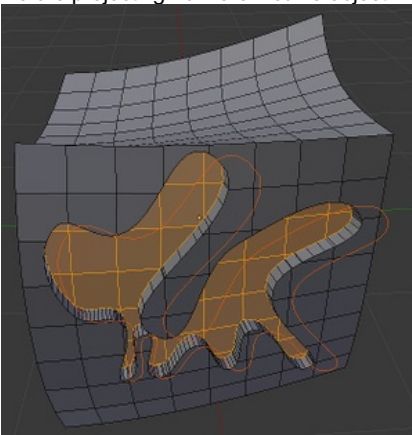
Before projecting from a mesh object



Resulting knife projection (extruded after)



Before projecting from a 3D curve object



Resulting knife projection (extruded after)

## Known Issues

Cutting holes into single faces may fail, this is the same limitation as with the regular knife tool but more noticeable for text, this can be avoided by projecting onto more highly subdivided geometry.

## Řez napříč (Bisect)

Mód: Edit mode

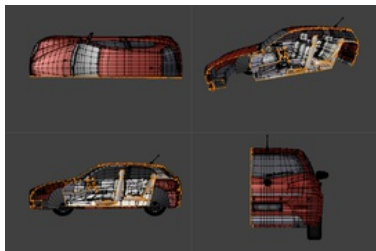
Menu: Mesh » Bisect

Nástroj "bisect" dává modeláři rychlou možnost rozříznout síťový model na dvě části pomocí parametricky definované roviny řezu. (nadpis kapitoly "Řez napříč" by měl správně být "Podélné, příčné a obecné řezy modelem").

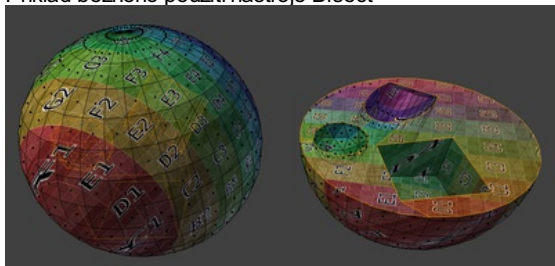
Je důležité poznamenat rozdíl mezi tímto způsobem řezu a řezem pomocí nože. Zde platí:

- Rovina řezu může být číselně upravována v operačním panelu.
- Řezy mohou odstranit geometrii jedné strany.
- Řezy mohou volitelně vyplnit vzniklé díry materiálem a UV povrchem a barvami založených na okolní geometrii.

Což znamená, že nástroj Bisect může vyřezávat části sítě bez jakéhokoli vytváření prázdných děr a prostorů.



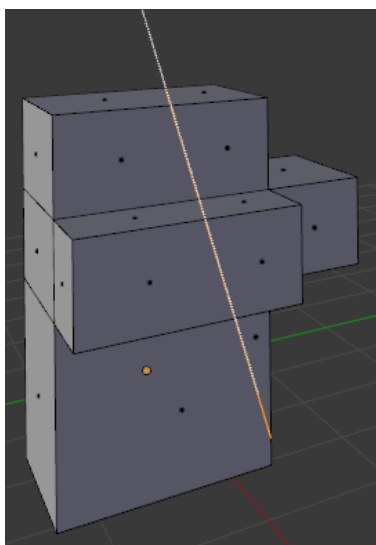
Příklad běžného použití nástroje Bisect



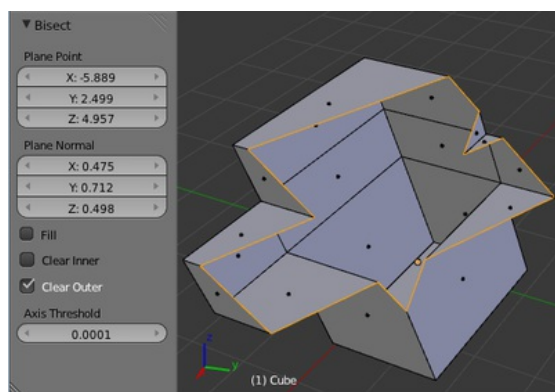
Příklad použití Bisect se zapnutou volbou výplně

## Postup

Model máme v editačním režimu (Edit-mode) a vybrané požadované prvky určené k řezu (zde pomocí A, nebo AA vybrán celý model). Pomocí menu Mesh-Bisect je kurzor přepnut do tvaru křížku a nyní nakreslíme úsečku reprezentující rovinu řezu podél pohledu 3D. PO této akci jsou v kontextovém menu zobrazeny parametry právě zvolené řezné roviny a volby viditelnosti jednotlivých částí modelu včetně stavu výplně plochy řezu. Akce je patrná z obrázku.



První krok - výběr roviny řezu.



Ukázka kontextového menu Bisect



### Poznámka

Po této akci je možné s rovinou dále pracovat, avšak musíme být stále v kontextu režimu editace téhož modelu. Přepnutí do Objektového módu způsobí zafixování řezu tak, jak jsme jej vytvořili, čili řezem vytvořené hrany a vrcholy se stávají plnohodnotnými součástmi řezaného modelu.

Spojení vrcholů

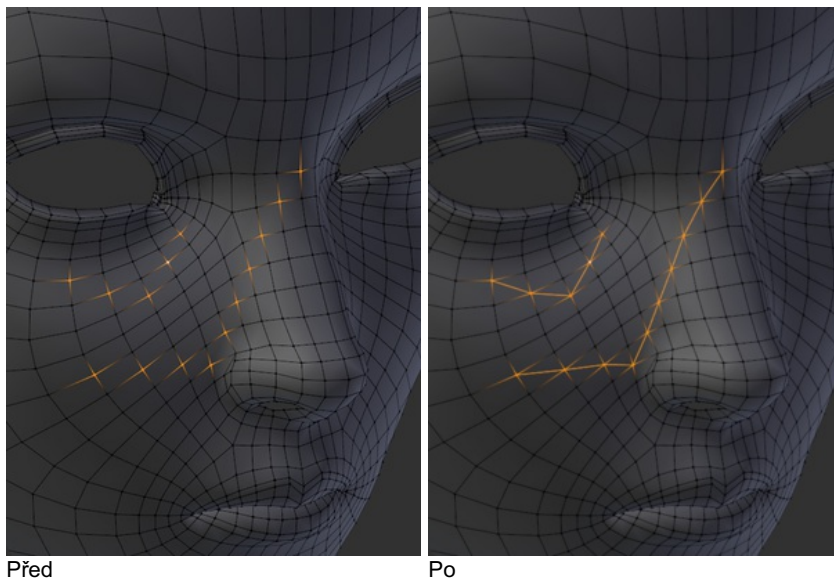
Mód: Edit režim

Klávesová zkratka: J

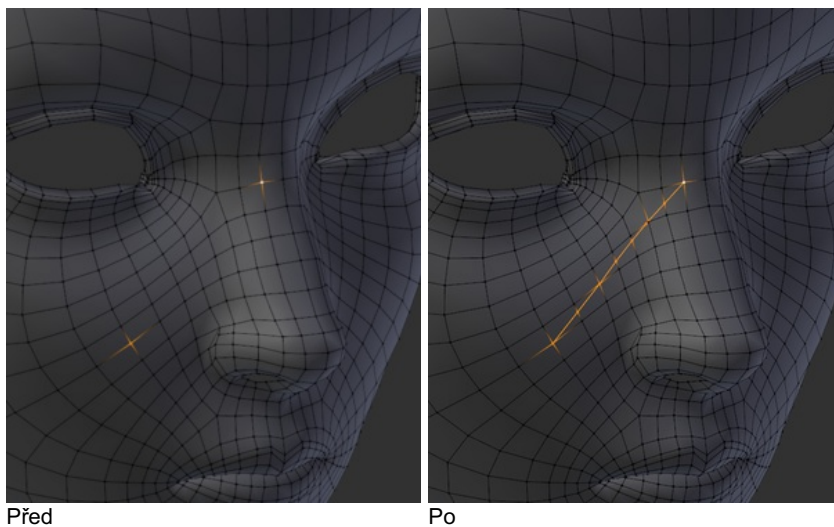
Menu: Mesh → Vertices → Connect

Tento nástroj spojí vybrané vrcholy pomocí nových hran. Hlavní rozdíl oproti vytváření hran je, že stěny jsou rozděleny nově vytvořenými hranami.

Pokud vyberete více vrcholů, povrchy budou rozděleny podle vybraných vrcholů.



Pokud vyberete pouze dva vrcholy, řez povrchem bude veden napříč nevybranými vrcholy podobně tak, jako při použití nástroje nože (knife) avšak s tím, že tato funkce řeže i nevybrané povrchy podle spojnice vybraných bodů..



Zkosení (Obroušení, Bevel)

Mód: Edit mode

Klávesová zkratka: CtrlB or W » Bevel

Menu: Mesh » Edges » Bevel nebo CtrlE » Bevel



Krychle se zkosenými hranami a bez obroušení.

The bevel tool allows you to create chamfered or rounded corners to geometry. A bevel is an effect that smooths out edges and corners. True world edges are very seldom exactly sharp. Not even a knife blade edge can be considered perfectly sharp. Most edges are intentionally beveled for mechanical and practical reasons.


Bevels are also useful for giving realism to non-organic models. In the real world, the blunt edges on objects catch the light and change the shading around the edges. This gives a solid, realistic look, as opposed to un-beveled objects which can look too perfect.

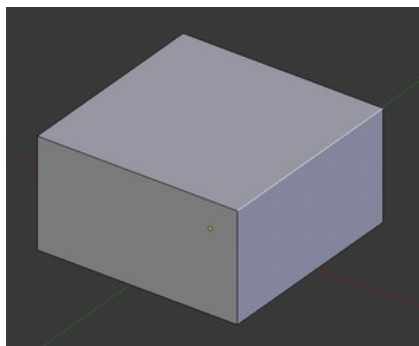
## Modifikátor zkosení

Modifikátor zkosení(viz také [Modifikátor Bevel](#)) je nedestruktivní alternativou nástroje obroušení. Dává uživateli stejné možnosti s výhodou, že zkosení může být navíc řízeno váhou vrcholů a s mnoha dalšími vylepšeními. Modifikátor Bevel nemá možnost rekurze, ale je možné jej vyvolat vícenásobně s dosti účinným efektem.

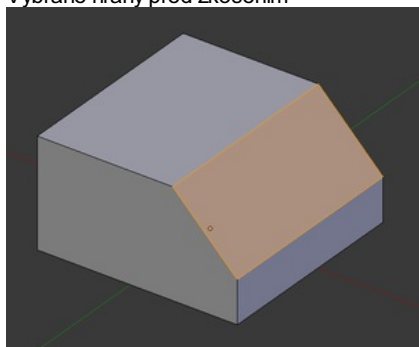
## Použití

The Bevel tool works only on selected edges. It will recognize any edges included in a vertex or face selection as well, and perform the bevel the same as if those edges were explicitly selected. The Bevel tool smooths the edges and/or “corners” (vertices) by “subdividing” them a specified number of times (see the options below for details about the bevel algorithm).

Use CtrlB or a method listed above to run the tool. Move the mouse to interactively specify the bevel offset, and scroll the Wheel  to increase or decrease the number of segments. (see below)

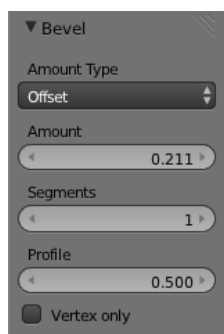


Vybrané hrany před zkosením



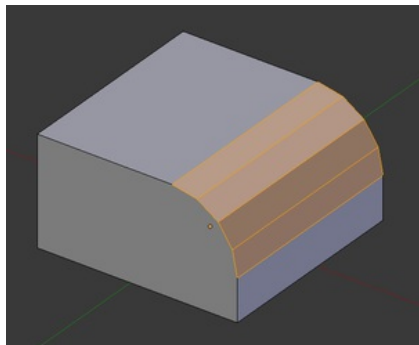
Výsledek zkosení

## Volby, nastavení



### Offset

You can change the bevel width by moving the mouse towards and away from the object, a bit like with transform tools. As usual, the scaling can be controlled to a finer degree by holding **⇧ Shift** to scale in 0.001 steps. **LMB** finalizes the operation, **RMB** or **Esc** aborts the action.



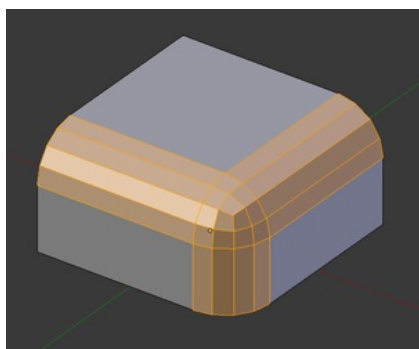
zkosení pomocí čtyř segmentů

### Segments

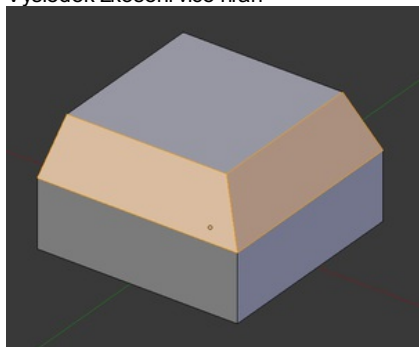
The number of segments in the bevel can be defined by scrolling the mouse **Wheel** to increase or decrease this value. The greater the number of recursions, the smoother the bevel.

Alternatively, you can manually enter a scaling value while using the tool, or in the Mesh Tool options panel after using the tool.

## Příklady



Výsledek zkosení více hran



Jiný příklad zkosení více hran

Různé editační nástroje

## Seřadit Prvky (Elements)

Tento nástroj (k dispozici v menu Specials, Vertices, Edges a Faces) umožňuje změnit pořadí vybraných síťových prvků, a to pomocí různých metod. Všimněte si, že při volání z nabídky Specials jsou dotčené typy prvků stejné jako právě vybrané.

View Z Axis (pohled podle osy Z)

Třídí podél osy Z aktivního pohledu. Od nejzažšího, po nejbližší. (anebo opačně pomocí volby Reverse).

View X Axis (pohled podle osy X)

Třídí podél osy X aktivního pohledu. Od leva doprava (anebo opačně pomocí volby Reverse).

Cursor Distance (vzdálenost kurzoru)

Třídí od nejzažšího k nejbližšímu dle vzdálenosti od 3D kurzoru (anebo opačně pomocí volby Reverse).

Material

**Pouze pro plochy!** Třídí plochy od těch, které mají nejnižší materiálový index k nevyššímu. Pořadí uvnitř "materiálových grup" zůstává beze změn. Volba opačného pořadí Reverse má vliv pouze na řazení materiálů, *nikoli* pořadí ploch uvnitř materiálů.

Selected (vybrané)

Přemístí všechyn vybrané prvky k začátku (nbo konci v případě volby Reverse) bez efektu jejich relativních pořadí. **Varování: tato volba má také vliv na indicie nevybraných prvků!**

Randomize (náhodné)

Náhodně indexuje vybrané provky (bez vlivu na nevybrané). Volba seed (semínka) umožňuje zvolit jinou náhodnost – při stejném počátečním parametru seed budou výsledky také stejné!

Reverse (opačně)

Duchaprostě obrátí pořadí vybraných prvků.

## Retopologie

Retopologie, anebo změna topologie - uspořádání prvků sítě modelu je proces modelovacího postupu, při kterém dochází k efektivnějšímu uspořádání hraničních vrcholů sítě tak, aby byl zachován původní tvar objektu.

Během modelování dochází k situaci, kdy jednotlivé části objektu nejsou z hlediska počtu a rozmístění vrcholů ideální. V některých oblastech může být síť příliš hustá, což je dáno detaily oblasti, a tím i neefektivní pro další matematické zpracování. Retopologie přeuspořádá vrcholy a hrany sítě modelu tak, aby byly z hlediska dalších modelovacích postupů ideální.

## Přilepování k síti

Zapnutím funkce přilepování a nastavením jeho způsobu se mohou některé vrcholy sítě projektovány na nejbližší povrch směrem od pohledu, tedy osy-Z.

Tato funkce umožňuje modelovat volně, bez koncentrace na formu a na detailní topologii objektu.

Viz [Přilepování](#)

## Modifikátor Shrinkwrap

Tento [Shrinkwrap modifikátor](#) je užitečný ve spojení s přilepováním na povrchy. Shrinkwrap je ekvivalentem pro smršťovací fólii, tedy pojem svou funkci vysvětluje. Pokud editujeme novou síť bez zapojení přilepování, modifikátor shrinkwrap bude umožňovat přilepit novou síť ku staré.

Supiny vrcholů (vertex Groups)

[- Create&Delete vertex groups](#)

General overview about creating and maintaining Vertex Groups

[- Weight Edit](#)

editing Vetex group weights

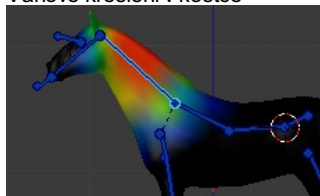
[- Weight Paint](#)

The weight Paint Mode

[- Weight Paint Tools](#)

the Weight Paint tools

## Váhové kreslení v kostce



- Do režimu váhového kreslení (Weight Paint) se dostanete pomocí menu Mode (hotkey=Ctrl⇧ Tab). Vybraný síťový objekt je zobrazen lehce vystínován barvami duhového spektra.
- Barva vizualizuje váhu náležící každému vrcholu aktivní skupiny vrcholů. Modrá znamená odvážené (lehké), červená těžké, plně vážené.
- Je možné upravit barvy pomocí váhového gradientu pomocí menu Custom Weight Paint Range v záložce System a User Preferences - uživatelské nastavení.
- Váhy vrcholů nakreslíte pomocí štětců. Začátek malování automaticky přidává váhu. (a new Vertex Group is created if needed).

**Užitečné klávesové zkratky**

Zkratky mohou zrychlit váhové kreslení, protože:

mění velikost štětce

F then drag to new brush size

vytváří lineární (přímkový) gradient

Alt LMB  then drag

vytváří radiální (kruhový) gradient

AltCtrl LMB  then drag

kreslí *Clipping Border*

AltB then drag the clipping border to select the part of the 3D window which shall be kept visible. You can then draw only in this part. Press AltB again to remove the *clipping border*.

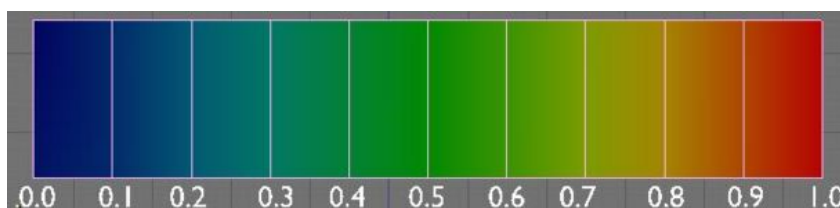
## Režim váhového kreslení

Vertex Groups can potentially have a very large number of associated vertices and thus a large number of weights (one weight per assigned vertex). Weight Painting is a method to maintain large amounts of weight information in a very intuitive way. It is primarily used for rigging meshes, where the vertex groups are used to define the relative bone influences on the mesh. But we use it also for controlling particle emission, hair density, many modifiers, shape keys, etc.

The basic principle of the method is: the weight information is literally painted on top of the Mesh body by using a set of Weight brushes. And since painting is always associated with color, we also need to define ...

### Barevné stupně váhy

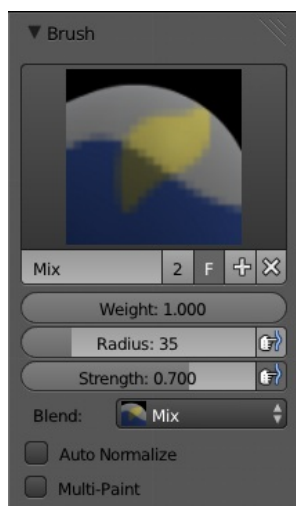
Weights are visualized by using a cold/hot color system, such that areas of low influence (with weights close to 0.0) are drawn in blue (cold) and areas of high influence (with weights close to 1.0) are drawn in red (hot). And all in-between influences are drawn in rainbow colors, depending on their value (blue, green, yellow, orange, red)



**Image 3:** The color spectrum and their respective weights.

In addition to the above described color code, Blender has added (as an option) a special visual notation for unreferenced vertices: They are drawn in black. Thus you can see the referenced areas (drawn in cold/hot colors) and the unreferenced areas (in black) at the same time. This is most practical when you look for weighting errors (we will get back to this later).

## Štětce



The Brush panel in the Tool Shelf

Painting needs paint brushes and Blender provides a Brush Panel within the Tool Shelf when it operates in Weight Paint Mode. You find predefined Brush Presets when you click on the large Brush Icon at the top of the brush Panel. And you can make your own presets as needed. See below for the available brush presets and to create custom presets.

#### Vlastnosti hlavního štětce

The most important and frequently modified properties are:

##### Weight

The weight (color) to be used by the brush. However, the weight value is applied to the Vertex Group in different ways depending on the selected Brush Blending mode (see below).

##### Strength

This is the amount of paint to be applied per brush stroke. What that means exactly also depends on the Brush Blending mode.

##### Radius

The radius defines the area of influence of the brush. Note: You can also change the Brush radius with a keyboard shortcut while painting. Just press F at any time, then drag the mouse to increase/reduce the brush radius. Finally click LMB to use the new setting. Or press the Esc key at any time to return to the current settings.

##### Blend mode

The brush Blending mode defines in which way the weight value is applied to the Vertex Group while painting. Blender provides 7 different Blending modes:

- **Mix:** In this Blend mode the Weight value defines the target weight that will eventually be reached when you paint long enough on the same location of the mesh. And the strength determines how many strokes you need to arrive at the target weight. Note that for strength = 1.0 the target weight is painted immediately, and for Weight = 0.0 the brush just does nothing.
- **Add:** In this blend mode the specified weight value is added to the vertex weights. The strength determines which fraction of the weight gets added per stroke. However, the brush will not paint weight values above 1.0.
- **Subtract:** In this blend mode the specified weight is subtracted from the vertex weights. The strength determines which fraction of the weight gets removed per stroke. However the brush will not paint weight values below 0.0.
- **Lighten:** In this blend mode the specified weight value is interpreted as the target weight very similar to the Mix Blend mode. But only weights below the target weight are affected. Weights above the target weight remain unchanged.
- **Darken:** This Blend mode is very similar to the Lighten Blend mode. But only weights above the target weight are affected. Weights below the target weight remain unchanged.
- **Multiply:** Multiplies the vertex weights with the specified weight value. This is somewhat like subtract, but the amount of removed weight is now dependent on the Weight value itself.
- **Blur:** tries to smooth out the weighting of adjacent vertices. In this mode the Weight Value is ignored. The strength defines how effectively the blur is applied.

#### Volby normalizace

Blender also provides Options regarding the automatic normalizing of all affected Vertex groups:

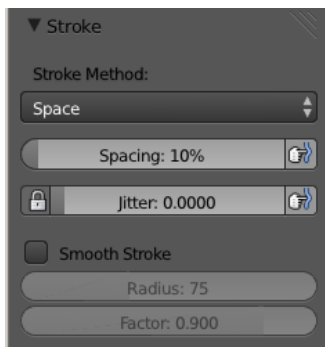
##### Auto Normalize

Ensures that all deforming vertex groups add up to 1 while painting. When this option is turned off, then all weights of a vertex can have any value between 0.0 and 1.0. However, when Vertex Groups are used as Deform Groups for character animation then Blender always interprets the weight values relative to each other. That is, Blender always does a normalization over all deform bones. Hence in practice it is not necessary to maintain a strict normalization and further normalizing weights should not affect animation at all.

##### Multi-Paint

Paint on all selected Vertex Groups simultaneously. This option is only useful in the context of Armatures, where you can select multiple Vertex Groups by selecting multiple Pose bones.

#### Definice tvaru čáry



Stroke Panel

## Stroke Method

- **Airbrush:** Keep applying paint effect while holding mouse down (spray)
- **Space:** Limit brush application to the distance specified by spacing (see below)
- **Dots:** Apply paint on each mouse move step

## Rate (only for Airbrush)

Interval between paints for airbrush

## Spacing (only for Space)

Limit brush application to the distance specified by spacing

## Jitter

Jitter the position of the brush while painting

## Smooth Stroke

Brush lags behind mouse and follows a smoother path

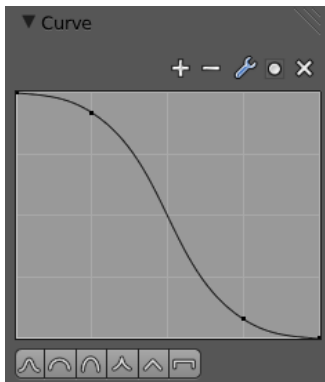
## Radius

Minimum distance from last point before stroke continues

## Factor

Higher values give a smoother stroke

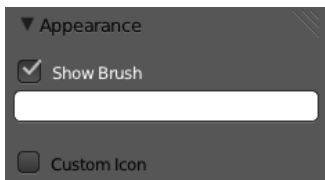
## Křivka přtlaku štětce



Curve Panel

The brush falloff editor allows you to specify the characteristics of your brushes to a large extent. The usage should be obvious and intuitive.

## Projevy štětce



Brush appearance

## Show Brush

makes the brush visible as a circle (on by default)

## Color setter

To define the color of the brush circle

## Custom icon

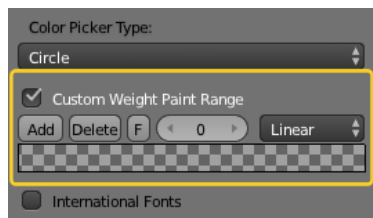
Allows definition of a custom brush icon

## Přednastavené štětce

Blender provides several Brush presets:

- **Mix, Draw, Brush:** uses the Mix Blending mode to draw the brush weight with varying strength and brush falloff
- **Add:** uses the Add Blending mode
- **Subtract:** uses the Subtract Blending mode
- **Lighten:** uses the Lighten Blending mode
- **Darken:** uses the Darken Blending mode
- **Multiply:** uses the Multiply Blending mode
- **Blur:** uses the Blur Blending mode

## Přednastavení barevného prostoru kreslení



Blender allows customization of the color range used for the Weight Paint colors. You can define the color band as you like; for example, you can make it purely black/white (similar to maya Weight painting), and you can even use Alpha values here.

You find the customizer in the User Properties section, in the System Tab.

## Selection Masking

If you have a complex mesh, it is sometimes not easy to paint on all vertices in Weight Paint mode. Suppose you only want to paint on a small area of the Mesh and keep the rest untouched. This is where selection masking comes into play. When this mode is enabled, a brush will only paint on the selected verts or faces. The option is available from the footer menu bar of the 3D viewport (see icons surrounded by the yellow frame):



You can choose between Face Selection masking (left icon) and Vertex selection masking (right icon).

Select mode has some advantages over the default Weight Paint mode:

1. The original mesh edges are drawn, even when modifiers are active.
2. You can select faces to restrict painting to the vertices of the selected faces.
3. Selecting tools include:

### Details about selecting

The following standard selection operations are supported:

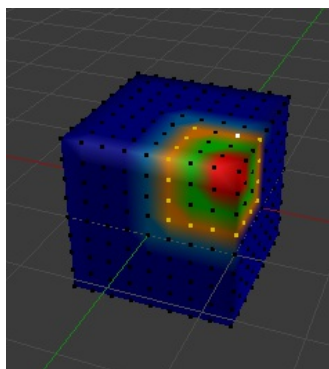
- RMB – Single faces. Use Shift RMB to select multiple.
- A – All faces, also to de-select.
- B – Block/Box selection.
- C – Select with brush.
- L – Pick linked (under the mouse cursor).
- CtrlL – Select linked.
- CtrlI – Invert selection (Inverse).



### Selecting Deform Groups

When you are doing weight painting for deform bones (with an Armature), you can select a deform group by selecting the corresponding bone. However, this Vertex Group selection mode is disabled when Selection Masking is active!

### Vertex Selection Masking

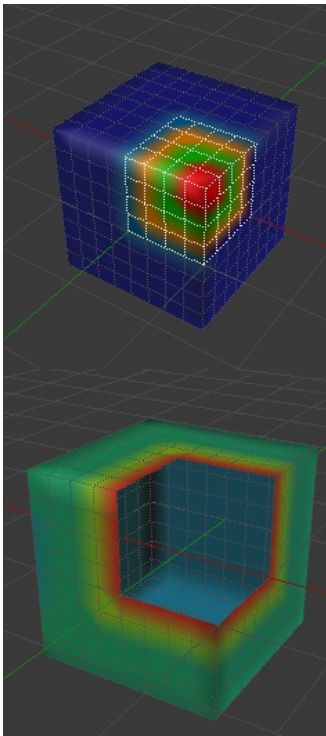


In this mode you can select one or more vertices and then paint only on the selection. All unselected vertices are protected from

unintentional changes.

Note: This option can also be toggled with the V key:

### Face Selection Masking



The Face Selection masking allows you to select faces and limit the weight paint tool to those faces, very similar to Vertex selection masking.

#### Hide/Unhide Faces

You also can hide selected faces as in Edit Mode with the keyboard Shortcut H, then paint on the remaining visible faces and finally unhide the hidden faces again by using AltH


#### Hide/Unhide Vertices

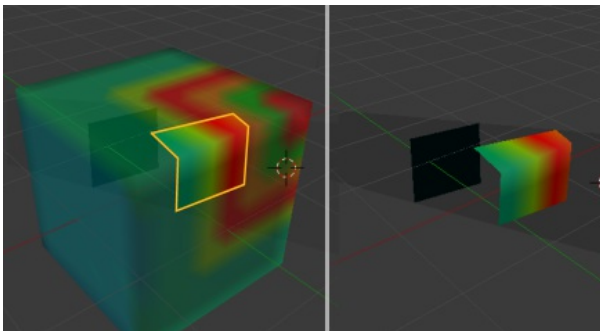
You cannot directly hide selected faces in vertex mask selection mode. However you can use a trick:

1. First go to Face selection mask mode
2. Select the areas you want to hide and then hide the faces (as explained above)
3. Switch back to Vertex Selection mask mode

Now the verts belonging to the hidden Faces will remain hidden.

#### The Clipping Border

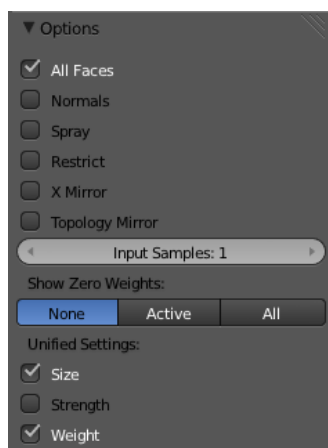
To constrain the paint area further you can use the *Clipping Border*. Press AltB and LMB -drag a rectangular area. The selected area will be "cut out" as the area of interest. The rest of the 3D window gets hidden.



You make the entire mesh visible again by pressing AltB a second time.

All weight paint tools that use the view respect this clipping, including border select, weight gradient and of course brush strokes.

## Weight Paint Options



The Weight Paint Options modify the overall brush behavior:

#### All faces

If this is turned off, you will only paint vertices which belong to a face on which the cursor is located. This is useful if you have a complicated mesh and you might paint on visually near faces that are actually quite distant in the mesh.

#### Normals

The vertex normal (helps) determine the extent of painting. This causes an effect as if painting with light.

#### Spray

The Spray option accumulates weights on every mouse move.

#### Restrict

The Restrict option limits the influence of painting to vertices belonging (even with weight 0) to the selected vertex group.

#### X-mirror

Use the X-mirror option for mirrored painting on groups that have symmetrical names, like with extension `.R/`, `.L`, or `_R/`, `_L`. If a group has no mirrored counterpart, it will paint symmetrically on the active group itself. You can read more about the naming convention in [Editing Armatures: Naming conventions](#). The convention for armatures/bones apply here as well.

#### Topology Mirror

Use topology-based mirroring, for when both side of a mesh have matching mirrored topology.

#### Input Samples

...

#### Show Zero Weights

- None
- Active
- All

**Unified Settings:** The Size, Strength and Weight of the brush can be set to be shared across different brushes, as opposed to per-brush.

- Spray: to constantly draw (opposed to drawing one stroke per mouse click).
- Restrict: to only paint on vertices which already are weighted in the active weight group. (No new weights are created; only existing weights are modified.)
- x-mirror: to draw symmetrically. Note the this only works when the character symmetry plane is z-y (character looks into y direction).
- Show Zero weights: To display unreferenced and zero weighted areas in black (by default).

## Weight Paint Tools



Blender provides a set of helper tools for Weight Painting. The tools are located in the weight tools panel.

The weight paint tools are full described in the [Weight Paint Tools](#) page


## Weight Painting for Bones


This is probably the most often used application of weight painting. When a bone moves, vertices around the joint should move as well, but just a little, to mimic the stretching of the skin around the joint. Use a "light" weight (10-40%) paint on the vertices around the joint so that they move a little when the bone rotates. While there are ways to automatically assign weights to an armature (see the

[Armature section](#)), you can do this manually. To do this from scratch, refer to the process below. To modify automatically assigned weights, jump into the middle of the process where noted:

- Create an armature.
- Create a mesh that will be deformed when the armature's bone(s) move.
- With the mesh selected, create an Armature modifier for your mesh (located in the Editing context, Modifiers panel). Enter the name of the armature.

*Pick up here for modifying automatically assigned weights.*

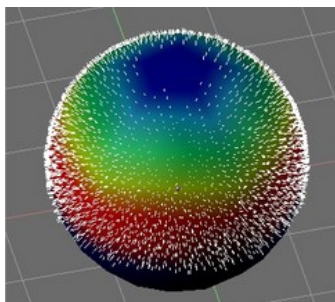
- Select the armature in 3D View, and bring the armature to **Pose mode** (Ctrl⇧ Tab, or the 3D View window header mode selector).
- Select a desired bone in the armature.
- Select your mesh (using RMB ) and change immediately to Weight Paint mode. The mesh will be colored according to the weight (degree) that the selected bone movement affects the mesh. Initially, it will be all blue (no effect).
- Weight paint to your heart's content. The mesh around the bone itself should be red (generally) and fade out through the rainbow to blue for vertices farther away from the bone.

You may select a different bone with RMB  while weight painting, provided the armature was left in Pose mode as described above. This will activate the vertex group sharing the name with the selected bone, and display related weights. If the mesh skins the bones, you will not be able to see the bones because the mesh is painted. If so, turn on X-Ray view (Buttons window, Editing context, Armature panel). While there on that panel, you can also change how the bones are displayed (Octahedron, Stick, B-Bone, or Envelope) and enable Draw Names to ensure the name of the selected bone matches up to the vertex group.

If you paint on the mesh, a vertex group is created for the bone. If you paint on vertices outside the group, the painted vertices are automatically added to the vertex group.

If you have a symmetrical mesh and a symmetrical armature you can use the option X-Mirror. Then the mirrored groups with the mirrored weights are automatically created.

## Weight Painting for Particles



Weight painted particle emission.

Faces or vertices with zero weight generate no particles. A weight of 0.1 will result in 10% of the amounts of particles. This option "conserves" the total indicated number of particles, adjusting the distributions so that the proper weights are achieved while using the actual number of particles called for. Use this to make portions of your mesh hairier than others by weight painting a vertex group, and then calling out the name of the vertex group in the VGroup: field (Particles panel, Object context).

Poznámka - v tomto i v dalších textech je uváděn pojem váha, který zdé má fyzikální význam hmotnosti, přestože správně váha tělesa je v podstatě jeho tíha, která je násobkem hmotnosti a tíhového zrychlení. Proto kupříkladu by měla být váha jednoho kilogramu hmoty menší v jiném tíhovém poli.

Tedy pokud fyzikové - puristé odpustí, bude pojmem váha myšlen pojem hmotnost a vice versa.

Nátroje vah



Blender provides a set of helper tools for Weight Painting. The tools are accessible from the Tool Shelf in Weight Paint mode. And they are located in the weight tools panel.

#### Volba podmnožiny (The Subset Option)

Some of the tools also provide a Subset parameter (in the Operator panel, displayed after the tool is called) with following options:

- Active Group
- Selected Pose Bones
- Deform pose Bones
- All Groups

All tools also work with Vertex Selection Masking and Face Selection masking. In these modes the tools operate only on selected verts or faces.



#### About the Blend tool

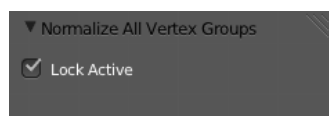
The Blend tool only works when "Vertex selection masking for painting" is enabled. Otherwise the tool button is grayed out.

## Normalize All

### Normalizace všech

For each vertex, this tool makes sure that the sum of the weights across all Vertex Groups is equal to 1. This tool normalizes all of the vertex groups, except for locked groups, which keep their weight values untouched.

#### Parametry operátoru

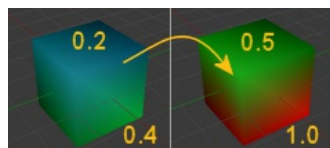


#### Lock Active

Keep the values of the active group while normalizing all the others.

Please note: Currently this tool normalizes ALL vertex groups except the locked vertex groups.

### Normalizace (Normalize)

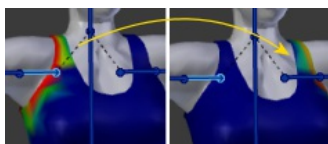


This tool only works on the active Vertex Group. All vertices keep their relative weights, but the entire set of weights is scaled up such that the highest weight value is 1.0

#### Parametry operátoru

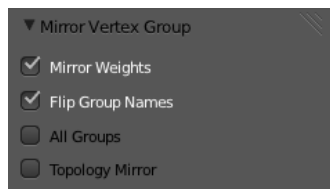
None

### Zrcadlení (Mirror)



This tool mirrors the weights from one side of the mesh to the opposite side (only mirroring along x-axis is supported). But note, the weights are not transferred to the corresponding opposite bone weight group. The mirror only takes place within the selected Vertex Group.

#### Parametry operátoru



#### Mirror Weights

Mirrors the weights of the active group to the other side. Note, this only affects the active weight group.

#### Flip Group Names

Exchange the names of left and right side. This option only renames the groups.

#### All Groups

Operate on all selected bones.

#### Topology Mirror

Mirror for meshes which are not 100% symmetric (approximate mirror).

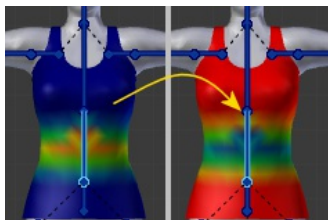


#### Mirror to opposite bone

If you want to create a mirrored weight group for the opposite bone (of a symmetric character), then you can do this:

- Delete the target Vertex Group (where the mirrored weights will be placed)
- Create a copy of the source bone Vertex Group (the group containing the weights which you want to copy)
- Rename the new Vertex Group to the name of the target Vertex Group (the group you deleted above)
- Select the Target Vertex Group and call the Mirror tool (use only the Mirror weights option and optionally Topology Mirror if your mesh is not symmetric)

## Inverze (Invert)



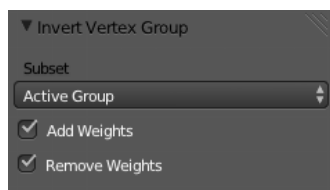
Replaces each Weight of the selected weight group by  $1.0 - \text{weight}$ .

Examples:

- original 1.0 converts to 0.0
- original 0.5 remains 0.5
- original 0.0 converts to 1.0

Note: Please see how the parameter settings change the behavior.

#### Parametry operátoru



#### Subset

Restrict the tool to a subset. See above (The Subset Option) about how subsets are defined.

#### Add Weights

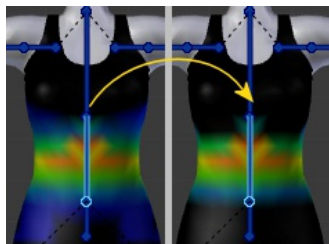
Add verts that have no weight before inverting (these weights will all be set to 1.0)

#### Remove Weights

Remove verts from the Vertex Group if they are 0.0 after inverting.

Note: Locked vertex Groups are not affected.

## Vyčištění (Clean)

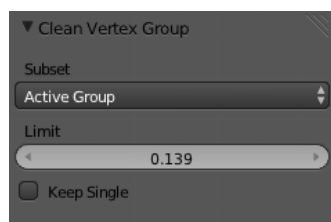


Removes weights below a given threshold. This tool is useful for clearing your weight groups of very low (or zero-) weights.

In the example shown, I used a cutoff value of 0.139 (see operator options below) so all blue parts (left side) are cleaned out (right side).

Note, the images use the Show Zero weights=Active option so that unreferenced Weights are shown in Black.

### Parametry operátoru



#### Subset

Restrict the tool to a subset. See above (The Subset Option) for how subsets are defined.

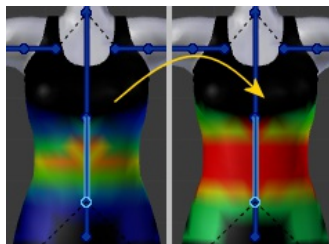
#### Limit

This is the minimum weight value that will be kept in the Group. Weights below this value will be removed from the group.

#### Keep Single

Ensure that the Clean tool will not create completely unreferenced verts (verts which are not assigned to any Vertex Group), so each vertex will keep at least one weight, even if it is below the limit value!

## Úrovně (Levels)



Adds an offset and a scale to all weights of the selected Weight Groups. with this tool you can raise or lower the overall “heat” of the weight group.

Note: No weight will ever be set to values above 1.0 or below 0.0 regardless of the settings.

### Parametry operátoru



#### Subset

Restrict the tool to a subset. See above (The Subset Option) for how subsets are defined.

#### Offset

A value from the range [-1.0, 1.0]) to be added to all weights in the Vertex Group.

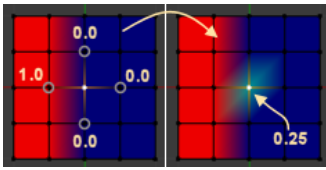
#### Gain

All weights in the Subset are multiplied with the gain. The drag sliders of this value allow only a range of [-10.0, 10.0]. However, you can enter any factor you like here by typing from the keyboard.

Note: Whichever Gain and Offset you choose, in all cases the final value of each weight will be clamped to the range [0.0, 1.0]. So you will never get negative weights or overheated areas (weight > 1.0) with this tool.

## Promíchání (Blend)

Blends the weights of selected vertices with adjacent unselected vertices. This tool only works in vertex select mode.

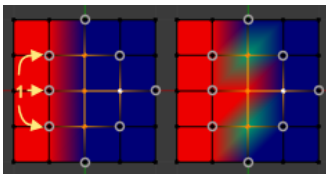


To understand what the tool really does, let's take a look at a simple example. The selected vertex is connected to 4 adjacent vertices (marked with a gray circle in the image). All adjacent vertices are unselected. Now the tool calculates the average weight of all connected **and** unselected verts. In the example this is

$$(1 + 0 + 0 + 0) / 4 = 0.25$$

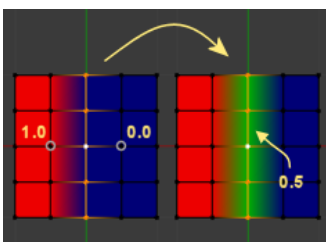
This value is multiplied by the factor given in the Operator parameters (see below).

- If the factor is 0.0 then actually nothing happens at all and the vertex just keeps its value.
- If the factor is 1.0 then the calculated average weight is taken (0.25 here).
- Dragging the factor from 0 to 1 gradually changes from the old value to the calculated average.



Now let's see what happens when we select all but one of the neighbors of the selected vert as well. Again all connected and unselected verts are marked with a gray circle. When we call the Blend tool now and set the Factor to 1.0, then we see different results for each of the selected verts:

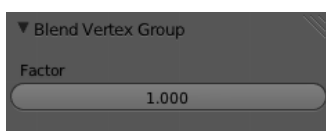
- The topmost and bottommost selected verts: are surrounded by 3 unselected verts, with an average weight of  $(1 + 0 + 0) / 3 = 0.333$ . So their color has changed to light green.
- The middle vertex: is connected to one unselected vert with weight = 1. So the average weight is 1.0 in this case, thus the selected vert color has changed to red.
- The right vert: is surrounded by 3 unselected verts with average weight =  $(0+0+0) / 3 = 0$ . So the average weight is 0, thus the selected vert color has not changed at all (it was already blue before blend was applied).



Finally let's look at a practical example (and explain why this tool is named Blend). In this example I have selected the middle edge loop. And I want to use this edge loop for blending the left side to the right side of the area.

- All selected vertices have 2 unselected adjacent verts.
- The average weight of the unselected verts is  $(1 + 0) / 2 = 0.5$
- Thus when the Blend Factor is set to 1.0 then the edge loop turns to green and finally does blend the cold side (right) to the hot side (left).

### Parametry operátoru



### Factor

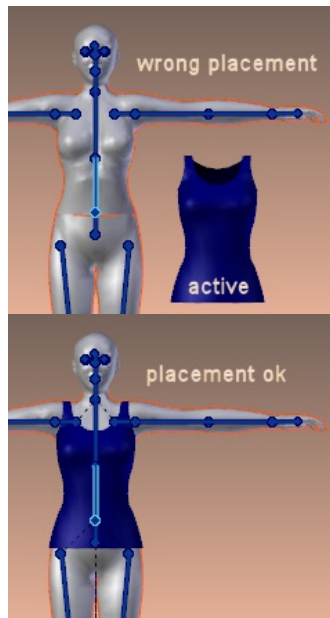
The effective amount of blending (range [0.0, 1.0]). When Factor is set to 0.0 then the Blend tool does not do anything. For

Factor > 0 the weights of the affected vertices gradually shift from their original value towards the average weight of all connected **and** unselected verts (see examples above).

## Přenos vah (Transfer Weights)

Copy weights from other objects to the vertex groups of the active Object. By default this tool copies all vertex groups contained in the selected objects to the target object. However you can change the tool's behavior in the operator redo panel (see below).

### Příprava kopie (Prepare the copy)



You first select all source objects, and finally the target object (the target object must be the active object).

It is important that the source objects and the target object are at the same location. If they are placed side by side, then the weight transfer won't work. You can place the objects on different layers, but you have to ensure that all objects are visible when you call the tool.

Now ensure that the Target Object is in Weight Paint mode.

### Spuštění nástroje (Call the tool)

Open the Tool Shelf and locate the Weight Tools panel. From there call the "Transfer weights" tool. The tool will initially copy all vertex groups from the source objects. However the tool also has an operator redo panel (which appears at the bottom of the tool shelf). From the redo panel you can change the parameters to meet your needs. (The available Operator parameters are documented below.)

### Redo Panel Confusion

You may notice that the Operator Redo Panel (see below) stays available after the weight transfer is done. The panel only disappears when you call another Operator that has its own redo Panel. This can lead to confusion when you use Transfer weights repeatedly after you changed your vertex groups. If you then use the still-visible redo panel, then Blender will reset your work to its state right before you initially called the Transfer Weights tool.

#### Obezlička (Workaround)

When you want to call the Transfer Weights tool again after you made some changes to your vertex groups, then always use the "Transfer Weights" Button, even if the operator panel is still available. Unless you really want to reset your changes to the initial call of the tool.

#### Parametry operátoru

Defaults are marked in boldface:



#### Group

- **Active:** Only copy to the Active Group in the active Object. This option only works when the active Object has an active Vertex Group set. Otherwise the Weight transfer will not do anything.
- **All:** Copy all Vertex groups from the selected objects to the Active Object.

#### Method

- **Nearest vertex in face:** to be documented
- **Nearest Face:** to be documented
- **Nearest vertex:** to be documented
- **Vertex Index** (verbatim copy, works only for meshes with identical index count)

#### Replace

- **Empty:** Only copy a weight to the active object if the vertex has not yet had a weight set in the group.
- **All:** delete all previous content of the target vertex group before copying the group from the source object.



#### Caveat!

If a vertex group is contained in 2 or more of the selected objects, then the result depends on the order in which the selected objects are processed. However, the order of processing cannot be influenced.

## Omezení počtu (Limit total)

Reduce the number of weight groups per vertex to the specified Limit. The tool removes lowest weights first until the limit is reached.

Hint: The tool can only work reasonably when more than one weight group is selected.

#### Parametry operátoru

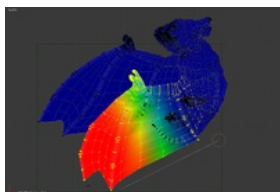
##### Subset

Restrict the tool to a subset. See above (The Subset Option) for how subsets are defined.

##### Limit

Maximum number of weights allowed on each vertex (default:4)

## Gradient váhy (Weight Gradient wip)



example of the gradient tool being used with selected vertices.

This is an interactive tool for applying a linear/radial weight gradient; this is useful at times when painting gradual changes in weight becomes difficult.

The gradient tool can be accessed from the Toolbar as a key shortcut:

- Linear: Alt LMB  and drag
- Radial: AltCtrl LMB  and drag

The following weight paint options are used to control the gradient:

- **Weight** - The gradient starts at the current selected weight value, blending out to nothing.
- **Strength** - Lower values can be used so the gradient mixes in with the existing weights (just like with the brush).
- **Curve** - The brush falloff curve applies to the gradient too, so you can use this to adjust the blending.

Blends the weights of selected vertices with unselected vertices. Hint: this tool only works in vertex select mode.

#### Parametry operátoru

#### Type

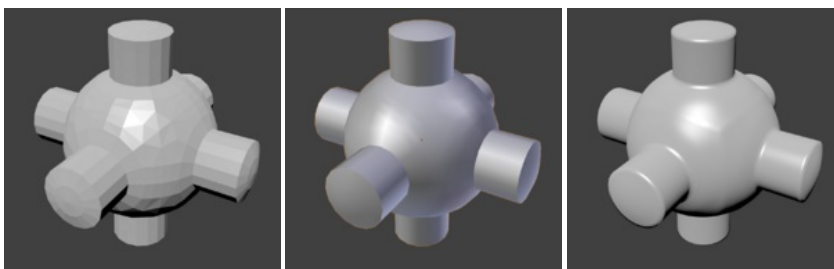
- Linear
- Radial

X Start  
X End  
Y Start  
Y End

Vyhlašování

TBD. 3.1.2014

## Mesh Shading



Example mesh rendered flat, smoothed using edge split, and using Subdivision Surface. Note how edges are rendered differently. [Sample .blend](#)

As seen in the previous sections, polygons are central to Blender. Most objects are represented by polygons and truly curved objects are often approximated by polygon meshes. When rendering images, you may notice that these polygons appear as a series of small, flat faces.

Sometimes this is a desirable effect, but usually we want our objects to look nice and smooth. This section shows you how to visually smooth an object, and how to apply the Auto Smooth filter to quickly and easily combine smooth and faceted polygons in the same object.

The last section on this page shows possibilities for smoothing a mesh's geometry, not only its appearance.

### Smooth shading

Mode: Edit and Object mode

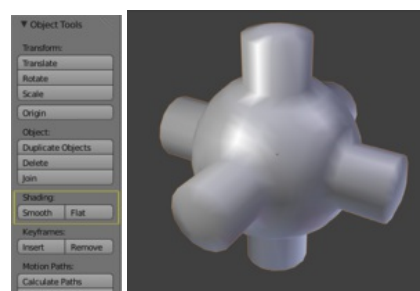
Panel: Mesh Tools (Editing context)

Hotkey: CtrlF » Shade Smooth / Shade Flat

Menu: Mesh » Faces » Shade Smooth / Shade Flat

The easiest way is to set an entire object as smooth or faceted by selecting a mesh object, and in Object mode, click Smooth in the Tool Shelf. This button does not stay pressed; it forces the assignment of the "smoothing" attribute to each face in the mesh, including when you add or delete geometry.

Notice that the outline of the object is still strongly faceted. Activating the smoothing features doesn't actually modify the object's geometry; it changes the way the shading is calculated across the surfaces, giving the illusion of a smooth surface. Click the Flat button in the Tool Shelf's Shading panel to revert the shading back to that shown in the first image above.



Same mesh smooth shaded

### Smoothing parts of a mesh

Alternatively, you can choose which edges to smooth by entering Edit mode, then selecting some faces and clicking the Smooth button. The selected edges are marked in yellow.

When the mesh is in Edit mode, only the selected edges will receive the "smoothing" attribute. You can set edges as flat (removing the "smoothing" attribute) in the same way by selecting edges and clicking the Flat button.

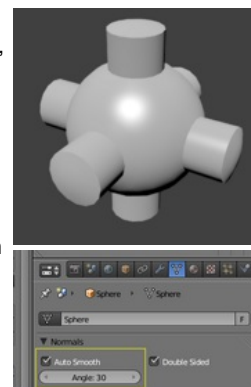
### Auto Smooth

Panel: Properties (Object Data context)

It can be difficult to create certain combinations of smooth and solid faces using the above techniques alone. Though there are workarounds (such as splitting off sets of faces by selecting them and pressing Y), there is an easier way to combine smooth and solid faces, by using Auto Smooth.

Auto smoothing can be enabled in the mesh's panel in the Properties window. Angles on the model that are smaller than the angle specified in the Angle button will be smoothed during rendering (i.e. not in the 3D view) when that part of the mesh is set to smooth. Higher values will produce smoother faces, while the lowest setting will look identical to a mesh that has been set completely solid.

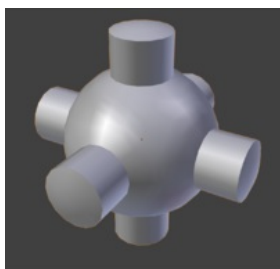
Note that a mesh, or any faces that have been set as Flat, will not change their shading when Auto Smooth is activated: this allows you extra control over which faces will be smoothed and which ones won't by overriding the decisions made by the Auto Smooth algorithm.



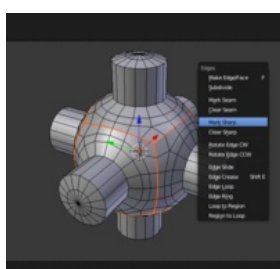
Example mesh with Auto Smooth enabled

## Edge Split Modifier

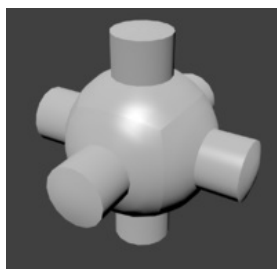
With the [Edge Split Modifier](#) we get a result similar to Auto Smooth with the ability to choose which edges should be split, based on angle—those marked as sharp.



Edge Split modifier enabled, based on angle



Edges marked as sharp



Resulting render with sharp edge weighting

## Smoothing the mesh geometry

The above techniques do not alter the mesh itself, only the way it is displayed and rendered. Instead of just making the mesh look like a smooth surface, you can also physically smooth the geometry of the mesh with these tools:

### Mesh editing tools

You can apply one of the following in Edit mode:

#### [Smooth](#)

This relaxes selected components, resulting in a smoother mesh.

#### [Laplacian Smooth](#)

Smooths geometry by offers controls for better preserving larger details.

#### [Subdivide Smooth](#)

Adjusting the smooth parameter after using the subdivide tool results in a more organic shape. This is similar to using the subdivide modifier.

#### [Bevel](#)

This Bevels selected edged, causing sharp edges to be flattened.

### Modifiers

Alternatively, you can smooth the mesh non-destructively with one or several of the following modifiers:

#### [Smooth Modifier](#)

Works like the Smooth tool in Edit mode; can be applied to specific parts of the mesh using vertex groups.

#### [Laplacian Smooth Modifier](#)

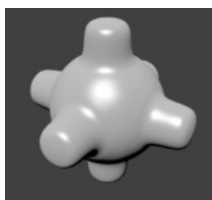
Works like the Laplacian Smooth tool in Edit mode; can be applied to specific parts of the mesh using vertex groups.

#### [Bevel Modifier](#)

Works like the Bevel tool in Edit mode; Bevel can be set to work on an angle threshold, or on edge weight values.

#### [Subdivision Surface Modifier](#)

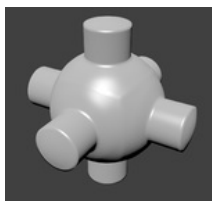
Catmull-Clark subdivision produces smooth results. Sharp edges can be defined with [subdivision creases](#) or by setting certain edges to “sharp” and adding an [EdgeSplit modifier](#) (set to From Marked As Sharp) before the Subsurf modifier.



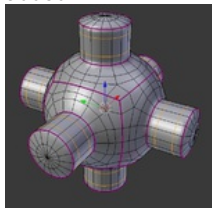
Subsurf



Using creased edges, and resulting subsurf artifacts



Extra edge loops  
added



3D view showing  
creased edges (pink)  
and added edges  
loops (yellow)

## Začištění sítě

Tyto nástroje slouží pro finální, nebo průběžné začištění případné poškozené geometrie a vyplnění chybějících ploch sítě.

## Vyplnění děr

Mód: Edit režim

Menu: Mesh » Clean up » Fill Holes

Nástroj pracuje s velkým výběrem a detekuje díry v povrchu sítě. Po té je vyplní stěnami..

Je rozdílný od nástroje vytváření stěn (creation operator) in v těchto tří aspektech:

- díry jsou detekovány, a tak není nutné manuálně vybírat hrany kolem děr.
- díry mohou převyšovat limit pro počet hran stěny (například jsou použity jako výplň pouze trojúhelníky a čtyřúhelníky).
- data sítě jsou kopírována včetně skrytých parametrů na pozadí (UV, barvy hrany všech vrstev) narozdíl od stejné, ale časově náročné ruční operace.

## Rozdělení (Split) nerovinných stěn

Mód: Edit režim

Menu: Mesh » Clean up » Split Non-Planar Faces

Tento nástroj odstraňuje geometrickou nejednoznačnost tím, že rozděluje nerovinné plochy, jejich prohnutí je nad daným limitem.

## Křivky



Logo ptáka vytvořené  
Bezierovými křivkami.

Křivky a [povrchy](#) jsou specifickým typem objektů Blenderu. Jejich popis je dán matematickými funkcemi a nejen sérií bodů. Blender nabízí obojí [bezierovy](#) křivky a křivky typu [NURBS](#) a povrchy. Zkratka NURBS pochází z anglického "Non-uniform rational basis spline", což je matematický model běžně používaný v počítačové grafice pro generování a reprezentování křivek a ploch, které nabízejí velkou flexibilitu a přesnost při manipulaci jak s analytickými tak s volnými tvary. Křivky jsou definovány pomocí skupiny řídicích bodů ("control vertices"), které tvoří řídicí polygon ("control polygon"). Velkou výhodou oproti polygonálním sítím je, že křivky potřebují k popisu mnohem méně údajů a mohou tak s využitím méně paměti a úložiště zajistit excelentní výsledky tvarů a celkově ušetřit čas při vytváření a modelování. Avšak na druhé straně může tento procedurální popis povrchů zvýšit čas renderování.

Určité modelovací techniky jako jsou [extruze profilu podél cesty](#) je možné uskutečnit pouze použitím křivek. Na druhé straně pokud používáme křivky na úrovni vrcholů je řízení velice obtížné a pokud je zapotřebí precizní kontrola [editace síťoviny](#) může být lepší volba editace.

Bezierovy křivky jsou nejpoužívanějším typem křivek v oblasti typografie při návrhu písma, loga. Také jsou hojně využívány v animacích jako [trasy](#) objektů pohybujících se podél a jako [F-křivky](#) pro změnu vlastností objektů jako funkce času.

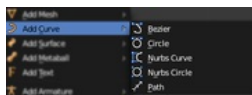
## Úvody

[Vytvoření loga Beziérovými křivkami »](#)

[Vytvoření povrchu dvěma nebo více křivkami »](#)

[Deformace sítě použitím křivek »](#)

## Křivková primitiva



Menu přidat křivku.

V menu Add Curve nabízí Blender pět základních primitiv křivek:

### Bezier Curve

Přidá otevřenou Beziérovu křivku s dvěma řídicími body.

### Bezier Circle

Přidá uzavřenou Beziérovu křivku ve tvaru 2D kruhu (vytvořenou ze čtyř řídicích bodů).

### NURBS Curve

Přidá otevřenou 2D křivku typu NURBS s pěti kontrolními body s uniformní (Uniform) uzly.

### NURBS Circle

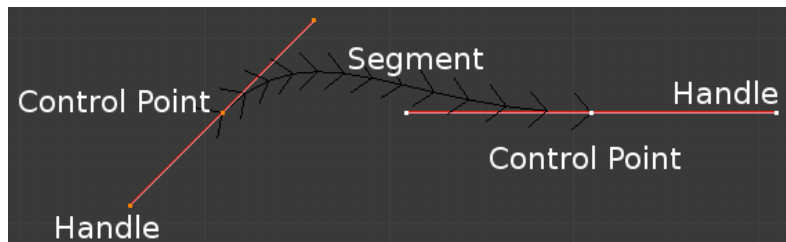
Přidá uzavřenou 2D křivku typu NURBS ve tvaru kruhu (vytvořenou z osmi řídicích bodů).

### Path

Přidá otevřenou 3D křivku typu NURBS vytvořenou pěti zarovnanými řídicími body s koncovým (Endpoint) uzlem a povolenou volbou CurvePath.

## Beziérové křivky

Hlavními prvky úpravy beziérových křivek jsou řídicí body (Control point) a uzly (Handles). Segment aktuální křivky se rozprostírá mezi dvěma řídicími body. Na obrázku níže je vidět řídicí bod uprostřed růžové úsečky zatímco uzly obsahují protažení od řídicího bodu. Běžně šipky podél segmentu reprezentují směr a současně **relativní** rychlost a směr pohybu objektu, který se případně podél křivky pohybuje. Toto může být změněno uživatelskou definicí parametru Speed lpo.



Beziérová křivka v editačním režimu.

## Úprava Beziérových křivek

Beziérové křivky mohou být upravovány přesunem řídicích bodů a uzlů..

1. Přidání křivky pomocí **⇧ Shift**A se nachází v menu Add a následnou položkou Curve » Bezier.
2. Stiskněte **⇧ Tab** pro vstup do editačního režimu.

3. Vyberte jeden řídící bod a přesuňte jej. Použijte zkratku LMB  pro potvrzení nového umístění, nebo RMB  pro zrušení.
4. Nyní vyberte jeden z uzlů a přesouvejte jej. Všimněte si jakých změn v křivosti je tím způsobováno.



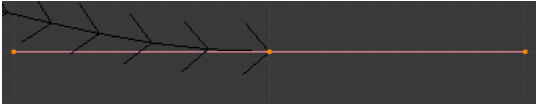

Pro přidání více řídících bodů

1. Vyberte alespoň dv ařídící body.
2. Stiskněte klávesu W a vyberte Subdivide (rozdělení).
3. Volitelně můžete stisknout F6 okamžitě po rozdělení pro změnu počtu dělení.

Je nutné poznamenat, že v editačním režimu nelze přímo vybrat segment. Pokud je to třeba, je nutné vybrat všechny řídící body tvořící segment kse kterým chcete manipulovat.

Existují čtyři typy uzlů Beziérových křivek. Mohou být k dipozici stiskem klávesy V a vyolbou se seznamu, nebo stiskem odpovídající kombinace horkých kláves. Uzly mohou být rotovány, přesouvány, škálovány a sraštěny, nebo roztaženy jako vrcholy u běžné síťoviny..

### Typy uzlů Beziérových křivek

Typ	Zkratka	Použití	Projev
Automatic	VA	This handle has a completely automatic length and direction which is set by Blender to ensure the smoothest result. These handles convert to Aligned handles when moved.	
Vector	VV	Both parts of a handle always point to the previous handle or the next handle which this allows you to create curves made of straight lines. Vector handles convert to Free handles when moved.	
Aligned	VL	These handles always lie in a straight line, and give a continuous curve without sharp angles.	
Free	VF	The handles are independent of each other.	

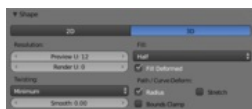
Additionally, the VT shortcut can be used to toggle between Free and Aligned handle types.

## Vlastnosti křivek

Vlastnosti křivek mohou být nastaveny pomocí volby Object Datav menu Properties Header (ukázáno níže modře).



### Tvar



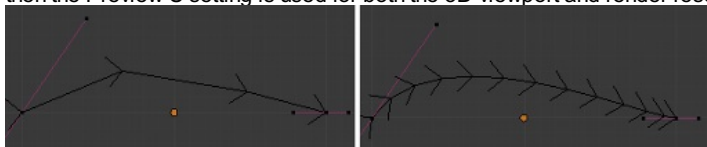
Curves Shape panel.

### 2D and 3D Curves

By default, new curves are set to be 3D which means that Control Points can be placed anywhere in 3D space. Curves can also be set to 2D which constrains the Control Points to the Curve's local XY axis.

### Resolution

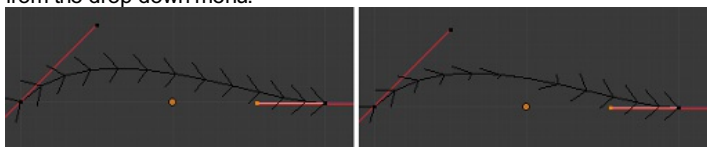
The *resolution* property defines the number of points that are computed between every pair of Control Points. Curves can be made more or less smoother by increasing and decreasing the resolution respectively. The Preview U setting determines the resolution in the 3D viewport while the Render U setting determines the Curve's render resolution. If Render U is set to zero (0), then the Preview U setting is used for both the 3D viewport and render resolution.



Curves with a resolution of 3 (left) and 12 (right).

### Twisting

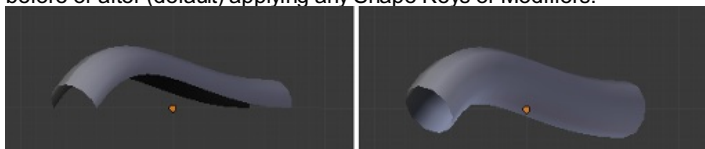
A 3D Curve has Control Points that are not located on the Curve's local XY plane. This gives the Curve a twist which can affect the Curve normals. You can alter how the twist of the Curve is calculated by choosing from Minimum, Tangent and Z-Up options from the drop down menu.



Curves with a twist of *minimum* (left) and *tangent* (right).

**Fill**

Fill determines the way a Curve is displayed when it is Beveled (see below for details on Beveling). When set to Half (the default) the Curve is displayed as half a cylinder. The Fill Deformed option allows you to indicate whether the Curve should be filled before or after (default) applying any Shape Keys or Modifiers.



Curves with a fill of *half* (left) and *full* (right).

**Path/Curve-Deform**

These options are primarily utilised when using a Curve as a Path or when using the Curve Deform property. The Radius, Stretch and Bounds Clamp options control how Objects use the Curve and are dealt with in more detail in the appropriate links below.

[Read more about Basic Curve Editing »](#)

[Read more about Paths »](#)

[Read more about Curve Deform »](#)

**Geometry**

Curves Geometry panel.

**Modification****Offset**

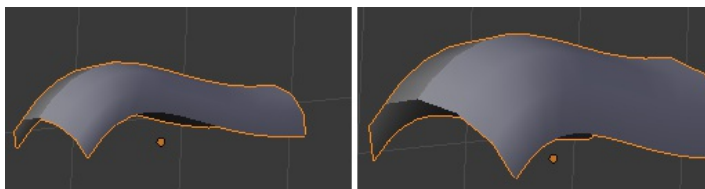
By default, text Objects are treated as curves. The Offset option will alter the space between letters.

**Extrude**

Will extrude the curve along both the positive and negative local Z axes.

**Bevel****Depth**

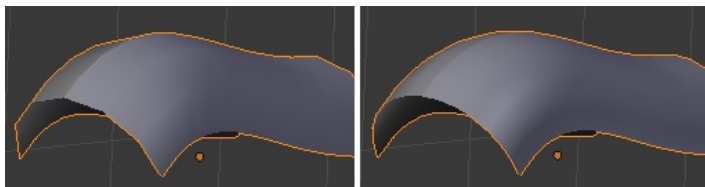
Changes the size of the bevel



A Curve with different Bevel depths applied.

**Resolution**

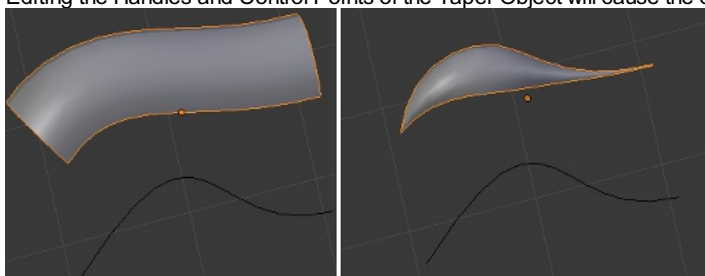
Alters the smoothness of the bevel



A Curve with different resolutions applied.

**Taper Object**

Tapering a Curve causes it to get thinner towards one end. You can also alter the proportions of the Taper throughout the tapered object by moving/scaling/rotating this Control Points of the Taper Object. The Taper Object can only be another Curve. Editing the Handles and Control Points of the Taper Object will cause the original object to change shape.

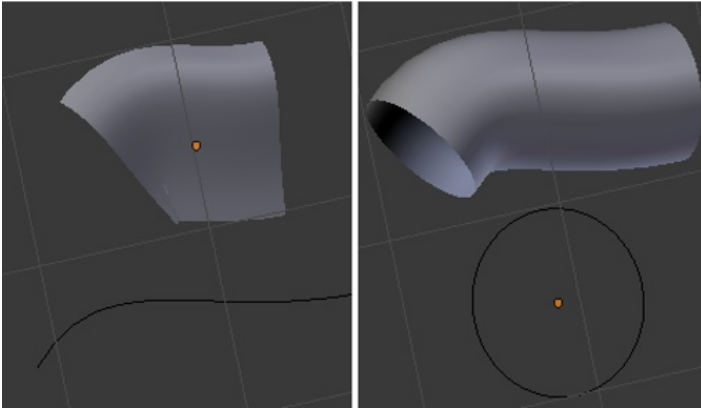


A Curve before (left) and after (right) a Bezier Curve Taper Object was

applied.

### Bevel Object

Beveling a Bezier Curve with a Bezier Curve as the Bevel Object generally gives it the appearance of a plane, while using a Bezier Circle as the Bevel Object will give it the appearance of a cylinder. The Bevel Object can only be another Curve. Editing the Handles and Control Points of the Bevel Object will cause the original object to change shape. Given the options available, it is best to experiment and see the results of this operation.



A Curve with the Bevel Object as a Bezier Curver (left) and as a Bezier Circle (right).

### Fill Caps

Seals the ends of a beveled Curve.

### Map Taper

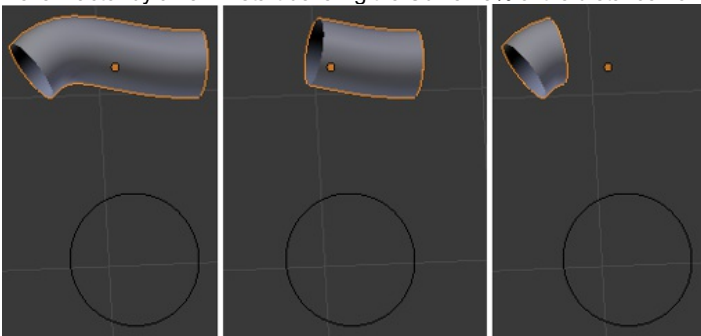
For Curves using a Taper Object and with modifications to the Start/End Bevel Factor the Map Taper option will apply the taper to the beveled part of the Curve (not the whole curve).



A Curve without (left) and with (right) Map Taper applied.

### Start Bevel Factor and End Bevel Factor

These options determine where to start the Bevel operation on the Curve being beveled. Increasing the Start Bevel Factor to 0.5 will start beveling the Curve 50% of the distance from the start of the Curve (in effect shortening the Curve). Decreasing the End Bevel Factor by 0.25 will start beveling the Curve 25% of the distance from the end of the Curve (again, shortening the Curve).



A Curve with no Bevel factor applied (left), with a 50% Start Bevel Factor (middle) and with a 25% End Bevel Factor (right).

[Read more about Advanced Curve Editing »](#)

### Path Animation

The Path Animation settings can be used to determine how Objects move along a certain path. See the link below for further information.

[Read more about utilizing Curves for paths during animation »](#)

### Active Spline



Curves Active Spline panel.

The Active Spline panel becomes available during Edit mode.

### Cyclic

Closes the curve.

### Resolution

Alters the smoothness of of each segment by changing the number of subdivisions.

## Interpolation

### Tilt

Alters how the tilt of a segment is calculated.

### Radius

Alters how the radius of a Beveled curve is calculated. The effects are easier to see after Shrinking/Fattening a control point AltS.

### Smooth

Smooths the normals of the Curve

## Non-Uniform Rational B-Splines (NURBS)

One of the major differences between Bezier Objects and NURBS Objects is that Bezier curves are approximations. For example, a Bezier circle is an *approximation* of a circle, whereas a NURBS circle is an *exact* circle. NURBS theory can be a *very* complicated topic. For an introduction, please consult the [Wikipedia page](#). In practice, many of the Bezier curve operations discussed above apply to NURBS curves in the same manner. The following text will concentrate only on those aspects that are unique to NURBS curves.

### Editing NURBS Curves

A NURBS Curve is edited by moving the location of the Control Points.

1. Place a Curve by ⇧ ShiftA to bring up the Add menu, followed by Curve » NURBS curve.
2. Press ⇧ Tab to enter Edit mode.
3. Select one of the Control Points and move it around. Use LMB to confirm the new location of the Control Point, or use RMB to cancel.
4. If you want to add additional Control Points, select both of them, press W and select Subdivide. Press F6 immediately after to determine how many subdivisions to make.

### Active Spline

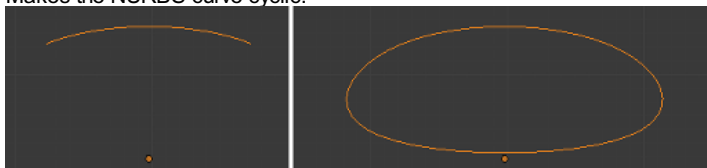


NURBS Active Spline panel.

One of the characteristics of a NURBS object is the *knot vector*. This is a sequence of numbers used to determine the influence of the control points on the curve. While you cannot edit the knot vectors directly, you can influence them through the Endpoint and Bezier options in the Active Spline panel. Note that the Endpoint and Bezier settings only apply to open NURBS curves.

### Cyclic

Makes the NURBS curve cyclic.



A NURBS curve with Cyclic applied.

### Bezier

Makes the NURBS curve act like a Bezier curve.

### Endpoint

Make the curve contact the end control points. Cyclic must be disabled for this option to work.



A NURBS curve with Endpoint enabled.

### Order

The order of the NURBS curve determines the area of influence of the control points over the curve. Higher order values means that a single control point has a greater influence over a greater relative proportion of the curve. The valid range of Order values is 2-6 depending on the number of control points present in the curve.



NURBS curves with orders of 2 (left), 4 (middle) and 6 (right).




## Path

As mentioned above, Curves are often used as [paths](#). Any curve can be used as a Path if the Path Animation option is selected.

The Path option available from the Add Curve menu is identical to a 3D NURBS curve, except that you do not have access to the Active Spline panel.

## Výběr křivky

Curve selection in Edit mode is much less complex than with meshes! Mainly this is because you have only one selectable element type, the control points (no select mode needed here...). These points are a bit more complex than simple vertices, however, especially for Béziers, as there is the central vertex, and its two handles...

The basic tools are the same as with [meshes](#), so you can select a simple control point with a LMB -click, add to current selection with  Shift LMB -clicks, Border-select, and so on.

One word about the Bézier control points: when you select the main central vertex, the two handles are automatically selected too, so you can grab it as a whole, without creating an angle in the curve. However, when you select a handle, only this vertex is selected, allowing you to modify this control vector...

L (or CtrlL) will add to the selection the cursor's nearest control point, and all the linked ones, i.e. all points belonging to the same curve. Note that for Bézier, using L with a handle selected will select the whole control point and all the linked ones.

## Menu pro výběr

With curves, all "advanced" selection options are regrouped in the Select menu of the 3D views header. Let's detail them.

Random...

Inverse

Select/Deselect All

Border Select

All these options have the same meaning and behavior as in [Object mode](#) (and the specifics of Border Select in Edit mode have already been discussed [here](#)).

## Každý n-tý

Mód: Edit režim

Klávesová zkratka: není

Menu: Select » Every Nth

This only works if you already have at least one control point selected. Using the current selection, it will add to it every nth control point, before and after the initial selection. The "selection step" is specified in the N pop-up numeric field shown during the tool start.

## Vybrat/odvybrat první/poslední

Mód: Edit režim

Klávesová zkratka: není

Menu: Select » Select/Deselect First, Select » Select/Deselect Last

These commands will toggle the selection of the first or last control point(s) of the curve(s) in the object. This is useful to quickly find the start of a curve (e.g. when using it as path...).

## Vybrat další/předchozí

Mód: Edit mode

Klávesová zkratka: není

Menu: Select » Select Next, Select » Select Previous

These commands will select the next or previous control point(s), based on the current selection (i.e. the control points following or preceding the selected ones along the curve).

## Více a méně

Mód: Edit režim

Klávesová zkratka: Ctrl+ NumPad/Ctrl- NumPad

Menu: Select » More/Less

These two options are complementary and similar to [those for meshes](#). Their purpose, based on the currently selected control points, is to reduce or enlarge this selection.

The algorithm is the same as with meshes, but results are more easy to understand:

- More: for each selected control point, select **all** its linked points (i.e. one or two...).
- Less: for each selected control point, if **all** points linked to this point are selected, keep this one selected. Otherwise, de-select it.

This implies two points:

- First, when **all** control points of a curve are selected, nothing will happen (as for Less, all linked points are always selected, and

of course, More can't add any). Conversely, the same goes when no control points are selected.

- Second, these tools will never “go outside” of a curve (they will never “jump” to another curve in the same object).

## Úpravy křivek

Tato stránka obsahuje základy úprav křivek. Složitější procedury, jako je extruze-roztažení, ohranění,taper jsou popsány v další kapitole:[Deformace křivek a protahování](#)

## Vytvoření křivek

Blender má základních pět odlišných primitiv pro křivky. Dva typy Bézier a tři NURBS:

Bezierova křivka

přidá otevřenou 2D Bézierovu křivku s dvěma kontrolními body.

Bezier Kruh

přidá uzavřenou 2D křivku kruhovitého tvaru (vytvořeno čtyřmi kontrolními body).

NURBS křivka

přidá otevřenou 2D NURBS křivku se čtyřmi kontrolními body a se stejnými Uniform úchyty.

NURBS kruh

přidá uzavřenou 2D NURBS křivku (vytvořenou osmi kontrolními body).

Path - cesta

přidá otevřenou 3D křivku NURBS vytvořenou pěti zarovnanými kontrolními body a s koncovým

(Endpoint) úchytem a s povoleným nastavováním cesty křivky (CurvePath).

## Zobrazení křivek

### Možnost i zobrazení

V editačním režimu jsou vlastnosti v menu Curve Display.

Handles

Vybírá zobrazení Beziérových úchytů v editačním režimu. Toto nemá vliv na projev

samotné křivky, viz [Curves](#), kde je popsáno více typů úchytů.

Normals

Vybírá zobrazení podle normál ([Curve Normals](#)).

Normal Size

Nastavuje zobrazení měřítka křivek podle normál.

(Not that this is relative to the control point scale which can be explicitly set, and is reflected in the normals as well.)

### Skryté prvky

Stejně jako v objektovém režimu je možné skryt vybrané části, což může být užitečné pro "vyčištění" pracovního prostoru kolem křivky v případě modelování prostředí s tisíce vrcholy...

Pro zakrytí použijte CtrlAltH, tlačítko skryt Hide v panelu Curve Tools1, anebo pomocí menu Curve » Show/Hide Control Points » Hide Selected (ukázat/schovat vybrané prvky).

Pro zobrazení skrytých prvků použijte AltH, tlačítko Reveal v nástrojovém panelu křivek Curve Tools1 nebo relevantní volbou ve stejném menu Curve » Show/Hide Control Points.

---

## Základní úpravy křivek (posuv, rotace, měřítko)

Mód: Edit mode

Klávesová zkratka: G/R/S

Menu: Curve » Transform » Grab/Move, Rotate, Scale, ...

Once you have a selection of one or more control points, you can grab/move (G), rotate (R) or scale (S) them, like many other things in Blender, as described in the [Manipulation in 3D Space](#) section.

Note that Bézier control points contain three vertices (the central one and the two handles), so a whole selected control point is equivalent to three selected vertices for transform tools (i.e. you can rotate and scale it, unlike standard mesh or NURBS vertices).

Note that in general, Bézier curves are easier to edit than NURBS, as when you modify a control point, you only affect the two curve segments on both side of the point. With NURBS, when you move a vertex, the curve can be modified up to three point on both side, depending on the order of the curve... Moreover, a Bézier curve always pass through the center of all its control points – NURBS are far from being so simple!

You also have in Edit mode an extra option when using these basic manipulations: the [proportional editing](#).

## Přichytávání

Mód: Edit mode

Panel: Curve Tools1 (Editing context)

[Mesh snapping](#) also works with curve components.

Both control points and their handles will be affected by snapping, except for within itself (other components of the active curve).

Snapping works with 2D curves, however points will be constrained to the curve's local XY axis.

## Nástroje deformace

Mód: Edit mode

Menu: Curve » Transform

The To Sphere, Shear, Wrap and Push/Pull transform tools are described in the [Manipulation in 3D](#) chapter.

The two other tools, Tilt and Shrink/Fatten Radius are related to [Curve Extrusion](#).

## Vyhlazování

Mód: Edit mode

Klávesová zkratka: W » smooth

Curve smoothing is available through the specials menu. For Bezier curves, this smoothing operation currently only smooths the positions of control points and not their tangents. End points are also constrained when smoothing.

## Zrcadlení

Mód: Edit mode

Klávesová zkratka: CtrlM

Menu: Curve » Mirror

The Mirror tool is also available, behaving exactly as with [mesh vertices](#).

---

## Set Bézier Handle Type

Mód: Edit mode

Panel: Curve Tools » Handles

Klávesová zkratka: V

Menu: Curve » Control Points » Set Handle Type


This only concerns Bézier curves. As we saw in a [previous page](#), these curves can have four types of handles (giving smooth or angled curve...). ⇧ ShiftH makes all selected control points automatic, H toggles between free and aligned, and V makes them vector. Follow above link for more details.


## Rozšiřování křivek

Mód: Edit mode

Klávesová zkratka: Ctrl LMB  or E

Menu: Curve » Extrude

Once a curve is created you can add new segments (in fact, new control points defining new segments...), either by extruding it, or placing new handles with Ctrl LMB  clicks. Each new segment is added to one end of the curve. A new segment will only be added if a single vertex, or handle, at one end of the curve is selected. If two or more control points are selected nothing is added (however, if you used the E Extrude command, all selected control points are placed in Grab mode...).

Note that unlike with meshes, you can't create a new curve inside the edited object by just Ctrl LMB -clicking with nothing selected – to do so, you can cut an existing curve in two parts (by [deleting a segment](#)), [copy](#) an existing one (⇧ ShiftD), or add a new one (Add menu)...

## Podrozdělení

Mód: Edit mode

Panel: Curve Tools (Editing context)

Klávesová zkratka: W » 1

Menu: Curve » Segments » Subdivide

Curve subdivision simply subdivides all selected segments by adding one or more control points between the selected segments.

Number of Cuts

Subdivides the segment multiple times evenly spacing this number of control points.

## Duplikace

Mód: Edit mode

Klávesová zkratka: ⇧ ShiftD

Menu: Curve » Duplicate

This command duplicates the selected control points, along with the curve segments implicitly selected (if any). The copy is selected and placed in Grab mode, so you can move it to another place.

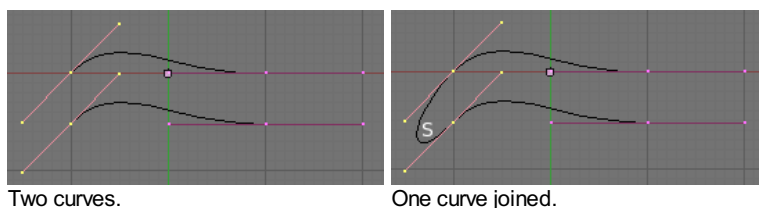
## Spojování segmetnů křivek

Mód: Edit mode

Klávesová zkratka: F

Menu: Curve » Make Segment

Two open curves can be combined into one by creating a segment between the two curves. To join two separated use one end control point from each curve. The two curves are joined by a segment to become a single curve.



Additionally, you can close a curve by joining the endpoints

Note that you can only join curves of the same type (i.e. Bézier with Bézier, NURBS with NURBS)

## Oddělení křivek

Mód: Edit mode

Klávesová zkratka: P

Menu: Curve » Separate

Curve object that are made of multiple distinct splines can be separated into their own objects. Note, if there is only one spline in a curve object, separating when something is selected will create a new curve object with no control points.

## Vymazání prvků

Mód: Edit mode

Klávesová zkratka: X or Del

Menu: Curve » Delete...

The Erase pop-up menu of curves offers you three options:

### Selected

This will delete the selected control points, *without* breaking the curve (i.e. the adjacent points will be directly linked, joined, once the intermediary ones are deleted). Remember that NURBS order cannot be higher than its number of control points, so it might decrease when you delete some control points... Of course, when only one point remains, there is no more visible curve, and when all points are deleted, the curve itself is deleted.

### Segment

This option is somewhat the opposite to the preceding one, as it will cut the curve, without removing any control point, by erasing one selected segment.

This option always removes *only one segment* (the last "selected" one), even when several are in the selection. So to delete all segments in your selection, you'll have to repetitively use the same erase option...

### All

As with meshes, this deletes everything in the object!

## Otevírání a uzavírání křivek

Mód: Edit mode

Klávesová zkratka: AltC

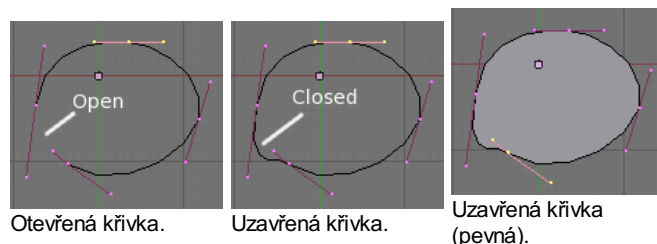
Menu: Curve » Toggle Cyclic

This toggles between an open curve and closed curve (Cyclic). Only curves with at least one selected control point will be closed/open.

The shape of the closing segment is based on the start and end handles for Bézier curves, and as usual on adjacent control points for NURBS. The only time a handle is adjusted after closing is if the handle is an Auto one. (*Open curve*) and (*Closed curve*) is the same Bézier curve open and closed.

This action only works on the original starting control-point or the last control-point added. Deleting a segment(s) doesn't change how the action applies; it still operates only on the starting and last control-points. This means that AltC may actually join two curves instead of closing a single curve!

Remember that when a 2D curve is closed, it creates a renderable flat face.



## Přepnutí směru

Mód: Edit mode

Klávesová zkratka: W » 2 NumPad

Menu: Curve » Segments » Switch Direction, Specials » Switch Direction

Tento příkaz "obráť" směr křivky s posledně vybraným elementem. (tj. startovní bod se stane koncovým a naopak). Nástroj je velice užitečný pokud používáme křivku jako zdrojovou cestu v případě modelování pásy atp...

## Nástroje pro konverzi

### Typy převodu křivky

Mód: Edit mode

Panel: Curve Tools1

Je možné převádět křivky typu spline do typů Beziérových, NURBS a Poly křivek.

Tlačítko Set Spline type (nastavení typu spline) ve skupině Curve na panelu Curve Tools umožňuje převádět vybrané spliny. Za poznámku stojí, že toto není "vychytralá" konverze, t.j. Beziérový tvar i.e. Blender se nepokouší vytvořit naprosto stejný tvar, ale pouze podle počtu řídících bodů... Například při konverzi NURBS do Beziérových křivek se každá skupina tří řídících prvků NURBS stává unikátní Beziérovými (centrální uzel a dva manipulátory).

### Převod křivky na síť

Mód: Object mode

Menu: Object » Convert to

Tento konverzní nástroj převádí objekt Curve na objekt typu Mesh se zachováním tvaru, obsahu a objemu v případě uzavřené křivky. Takto převedeným objektem je možné dále tvořit hrany a vrcholy...

### Převod sítě na křivku

Mód: Object mode


Menu: Object » Convert to

Síťové objekty, které se skládají ze propojených křivek mohou být převedeny na objekty typu křivka. Výsledná křivka bude typu poly, ale může být pro dosažení vyšší hladkosti převedena do více segmentů jak je popsáno výše.

### Rodičovství křivek

Mód: Edit mode

Klávesová zkratka: CtrlP

Pomocí dalších vybraných objektů [potomci](#) z jednoho nebo tří rodičovských kontrolních bodů (CtrlP je možné vytvořit rodiče. Po vybrání 1 nebo 3 kontrolních bodů stisknete Ctrl RMB  a jiného objektu pomocí CtrlP vytvoříte rodiče vrcholů.

## Háky

Mód: Edit mode

Klávesová zkratka: H

Menu: curve » control points » hooks

[Háky](#) mohou být přidány jako řídící body pro jeden nebo více bodů ostatních objektů.

## Nastavení cílové váhy

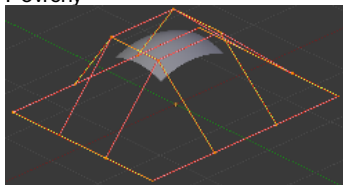
Mód: Edit mode

Menu: W » Set Goal Weight

Set Goal Weight

Toto nastavení "cílové váhy" vybraných kontrolních bodů má vliv na měkkost fyzikální charakteristiky. Křivka se pak snaží "přilepit" do svých původních pozic na základě své váhy.

## Povrchy



Povrch.

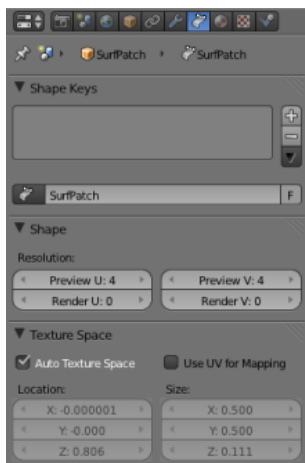
Křivky jsou objekty typu 2D a povrchy jsou jejich rozšířením do 3D. Blender však může pracovat pouze s povrchy typu NURBS, nikoli Bézierovými (jsou k dispozici Bezierovy uzly jak je vidět níže), nebo polygonálními (ale pro polygony jsou klasické sítě). Dokonce křivky a povrchy sdílejí stejný typ objektu (také s texty...), nejsou stejnou věcí, například v jednom objektu není možné využít křivky a povrchy.

Povrchy jsou rozměru 2D. Mají tedy dvě interpolační osy, U (pro křivky) a V. Je důležité porozumět, že *lze řídit interpolační pravidla (uzel, pořadí, rozlišení) nezávisle pro oba tyto rozměry* (kde položky (the U a V jsou k nastavení určeny).

Můžete se ptát, "ale povrch se jeví jako trojrozměrný, proč je tedy jen 2D?". Vlastností objektu 3D je, že má další parametr daný třetím rozměrem. Tím je objem, kdežto povrch, přestože může být uzavřený má pouze povrch o rozměru 2. Je nekonečně tenký. Proto pouze 2D objekty mají jen dvě interpolační rozměry, nebo osy, nebo rozměry. (pokud se trochu vyznáte v matematice, myslíte v neeuklidovské geometrii - nuže povrchy jsou pouze neeuklidovské roviny 2D). Pro klasický "příklad ze života" je možné srolovat kus papíru do válce, který sice ohraničuje určitý objem, ale stále jde o papír s obsahem několika čtverečních centimetrů. Tedy 2D objekt!

V podstatě jsou povrchy velice podobné výsledku [vytažení křivky](#) (by the way, I think it should be possible to convert an extruded curve in a surface, at least when only made of NURBS – but Blender cannot do it currently...).

## Nástroje úpravy povrchů



Surface Tools.

Panel Editing jsou stejné jako pro [křivky](#) pouze s menším počtem možností... A jako obvykle jsou k dispozici menu Select a Surface v záhlaví pohledu 3D a Specials (W) v menu pop-up.

## Vizualizace

There is nearly no difference with NURBS curves, except that the *u* direction is indicated by yellow grid lines, and the *v* one is materialized by pink grid lines, as you can see in (*Surface*).

You can [hide and reveal](#) control points just as with curves, and you have the same draw options in the [Curve Tools](#) panel.

## Struktura povrchu

Mnoho z konceptu [křivek](#), zvláště [NURBS](#) se vztahují k NURBS povrchům jako řídicí body, pořadí (Order), váhy (Weight), rozlišení (Resolution), atd. Následuje detailní popis rozdílů.

Je velice důležité porozumět rozdílu mezi křivkami NURBS a povrchy NURBS. První mají pouze jednu dimenzi, druhé mají dimenze dvě. Blender interně počítá povrchy NURBS a křivky NURBS úplně rozdílně. Je mnoho atributů, které lze oddělit, ale nejdůležitější je, že křivky NURBS mají jednu interpolační osu (U) a povrchy NURBS mají dvě interpolační osy (U a V).

However, you can have "2D" surfaces made of curves (using the [extrusion tools](#), or, to a lesser extent, the filling of closed 2D curves. And you can have "1D" curves made of surfaces, as a NURBS surface with only one row (either in U or V direction) of control points produces only a curve...

Visually you can tell which is which by entering Edit mode and looking at the 3D window's header: either the header shows "Surface" or "Curve" as one of the menu choices. Also, you can [extrude](#) a whole NURBS surface curve to create a surface, but you can't with a simple NURBS curve (we talk here about the "standard" Extrude tool, the one activated with the E shortcut, not the quite specific curve extrusion tools – yes, I know, it's not easy to follow...).

## Řídící body, řádky a mřížka (Grid)

Control points for NURBS surfaces are the same as for NURBS curves. However, their layout is quite constraining. The concept of

“segment” disappears, replaced by “rows” and the overall “grid”.

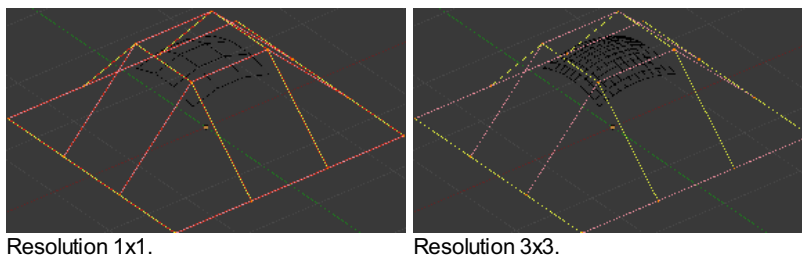
A “row” is a set of control points forming one “line” in one interpolation direction (a bit similar to [edge loops](#) for meshes). So you have “U-rows” and “V-rows” in a NURBS surface. The key point is that *all rows of a given type (U or V) have the same number of control points*. Each control point belongs to exactly one U-row and one V-row.

All this forms a “grid”, or “cage”, of which shape controls the shape of the NURBS surface. A bit like [lattice](#)...

This is very important to grasp: you cannot add a single control point to a NURBS surface, you have to add a whole U- or V-row at once (in practice, you will usually use the extrude tool, or perhaps the duplicate one, to add those...), containing exactly the same number of points as the others. This also means that you will only be able to “merge” different pieces of surfaces if at least one of their rows match together.

## Rozlišení povrchu

Just like [NURBS curves](#), Resolution controls the detail of the surface. The higher the Resolution the more detailed and smoother the surface is. The lower the Resolution the rougher the surface. However, here you have two resolution settings, one for each interpolation axis (U and V). Note that unlike with curves, you have only one resolution (the Resol U and V fields, in the Curve Tools panel)...



(*Resolution 1x1*) is an example of a surface resolution of 3 for both *u* and *v*. (*Resolution 3x3 surface*) is an example of a surface resolution of 12 for both *u* and *v*.



Resolution panel.

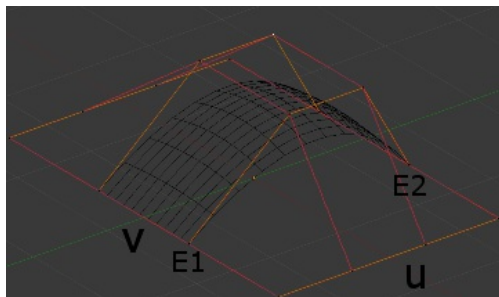
you can adjust the resolution for both preview and render, to don't slow things down in the viewport, but still get good render results.

## Uzavřené a otevřené povrchy

As curves, surfaces can be closed (cyclical) or opened, independently in both directions, allowing you to easily create a tube, donut or sphere shape, and they can be drawn as “solids” in Edit mode. This makes working with surfaces quite easy.

## Uzly (Knots)

Just like with [Křivky NURBS](#), NURBS surfaces have two knot vectors, one for each U and V axis. Here again, they can be one of Uniform, Endpoint, or Bezier, with same properties as for curves. And as with curves, only opened surfaces (in the relevant direction) are affected by this setting...



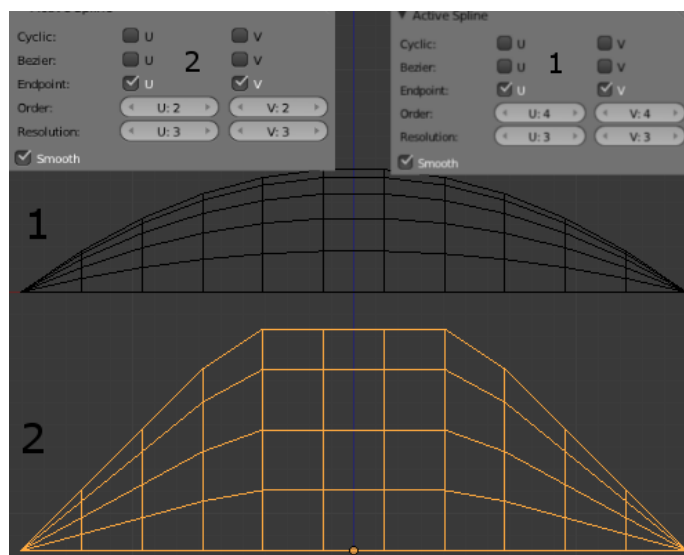
Endpoint U.

In (*Endpoint U*), the U interpolation axis is labeled as “u” and the V interpolation axis is labeled as “v”. The u’s interpolation axis has been set to Endpoint and as such the surface now extends to the outer edges from “E1” to “E2” along the *u* interpolation axis.

To cause the surface to extend to all edges you would set the *v*’s axis to Endpoint as well.

## Pořadí (Order)

One more time, this property is the same as with [NURBS Curves](#), it specifies how much the control-points are taken into account for calculating the curve of the surface shape. For high Orders, (1), the surface pulls away from the control-points creating a smoother surface – assuming that the [resolution](#) is high enough. For lowest Orders, (2), the surface follows the control-points, creating a surface that tends to follow the grid cage.

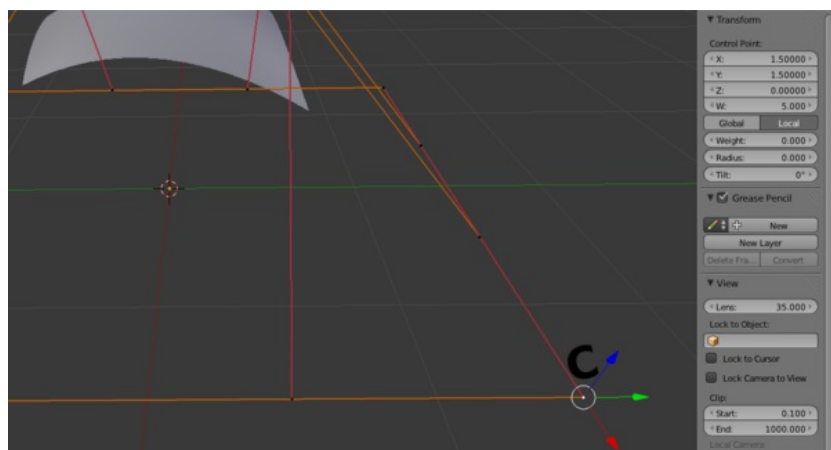


Order 2 and order 4 surface.

For illustration purposes, in both (*Order 4 surface*) and (*Order 2 surface*), the knot vectors were set to Endpoint causing the surface to extend to all edges.

You can set independently the order for each interpolation axis, and as for curves, it cannot be lower than **2**, and higher than **6** or the number of control points on the relevant axis.

## Váhy (Weight)



Pracují stejně jako s křivkami [NURBS](#)! Weight určují, jak mnoho řídící bod "vytáhne" křivku.

V (*váze povrchu* 5) samostatný řídící bod označený jako "C" má vlastní váhu (Weight) nastavenou na **5.0** zatímco ostatní na jejich přednastavenou hodnotu **1.0**. Jak je patrné řídící bod *vytahuje* povrch směrem k sobě.

Pokud všechny řídící body mají stejnou váhu (Weight), pak každý efektivně má vliv na každého. Je rozdíl ve vahách, které způsobují pohyb povrchu směrem k, nebo od řídícího bodu.

Váha (Weight) určitého řídícího bodu je patrná v panelu [Vlastnosti transformace](#) (N), v *poli W* (a nikoli v poli Weight field...).

## Přednastavené váhy

[File:NurbsSurfaceSphere.png?](#)

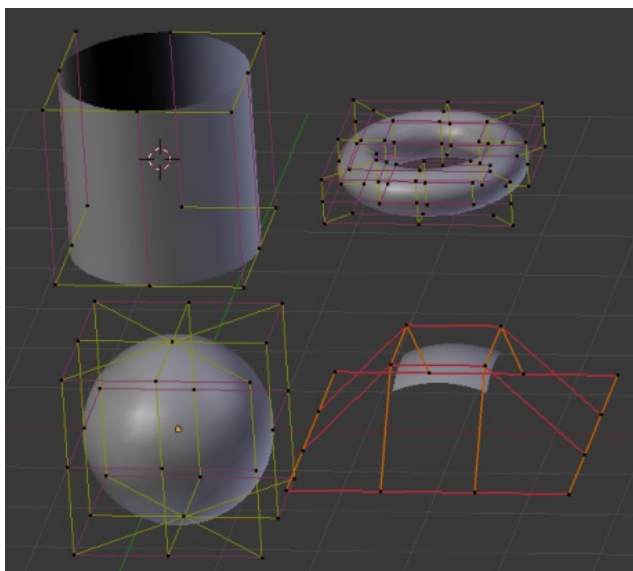
Povrch koule.

NURBS mohou vytvářet čisté tvary jako kruhy, válce a koule (pozn.: Bézierova kružnice není přesná kružnice). Pro vytvoření přesných kruhů, glóbů nebo válců musí být specifikovány určité váhy kontrolních bodů křivky – některé jsou přednastavené v nástrojích panelu Curve Tools (dolní pravý roh). Nastavení není zcela intuitivní a vyžaduje hlubší studium problematiky NURBS.

Viděli jsme v 1D [NURBS curves](#) jak vytvořit kružnici, podívejme se na to, jak vytvořit kouli pomocí 2D povrchů. Je to na stejném principu - pro vytvoření je třeba mít čtyři rozdílné váhy (**1.0**, **0.707** =  $\sqrt{0.5}$ , **0.354** =  $\sqrt{2}/4$ , and **0.25**) jsou čtyři přednastavené dostupné v panelu [Curve Tools](#)...

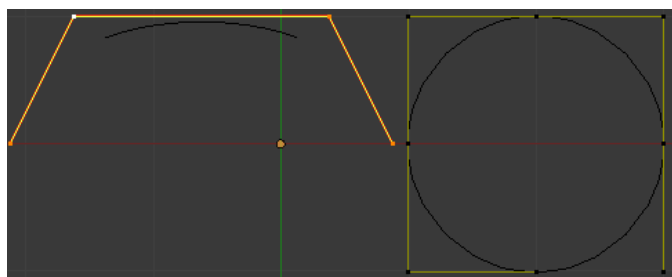
## Primitiva

Výpomocí pro tvorbu povrchů jsou k dispozici čtyři přednastavené NURBS povrchy. Nachází se v menu Add » Surface: NURBS Surface, NURBS Tube, NURBS Sphere a NURBS Torus.



Primitiva povrchu NURBS.

Existují dvě přednastavení povrchových křivek NURBS (pouze s jedním řídícím bodem na každý řádek V-row): NURBS Curve a NURBS Circle.



Primitiva křivky NURBS.

Kruhový povrch NURBS není nikdy vyplněn stejně jako jeho "reálný" protějšek ve skutečnosti...

Textové objekty

Mód: Edit mode (Text)

Panel: Curve and Surface, Font and Char (Editing context, F9)

Klávesová zkratka: F9

Menu: Add » Text



Příklady textu.

Text je objekt, který je přesně to, co znamená: text. Sdílí stejné objektové vlastnosti jako křivky a povrchy. Podporuje systémová písma (OpenType, TrueType, atd.), je vektorový, vytvořený z křivek (obecně Beziérových).

Blender uses a “Font System” to manage the mapping “letter codes › objects representing them in 3D views”. This implies that not only does the font system have its own *built-in* font, but it can use external fonts too, including *PostScript Type 1*, *OpenType* and *TrueType* fonts. And last but not least, it can use any objects existing in current .blend file as letters...

Texts in Blender allow you to create/render 2D or 3D texts, shaded as you want, with various advanced layout options (like justifying and frames), as we will see below. By default, letters are just flat filled surfaces, exactly as any closed 2D curve. But you can of course extrude them... And texts can follow other curves.

Of course, once you are happy with the shape of your text, you can convert it (with AltC, in Object mode), either to a curve, or directly to a mesh, allowing you to use on it all the powerful features of these objects...

(*Text Examples*) shows some examples of various fonts in action including the “blue” font that has been applied to a curve path.

## Notes

A maximum of **50000** characters is allowed per text object, however, be forewarned that the more characters a single text object has, the slower the object will respond interactively.

As you can see when you switch between Object and Edit modes, the Font panel remains the same. This means that its settings can be applied indifferently in both modes... and this implies that you cannot apply them to just a part of the mesh. So font, size, and so on, are common to all letters in a Text object. There is just one exception: the Bold/Italic buttons control properties specific to each letter (this is a way to use up to four different fonts in a text).

For optimum resource usage, only characters that are being used consume memory (rather than the entire character set).

## Editing Text

Mode: Edit mode

Hotkey: see below



Text in Edit mode.

Editing text is quite different than other object types in Blender, and happens mainly in two areas. First, the 3D view, of course, where you type your text, and have a few shortcuts, e.g. for applying [styles](#) – note however that most Blender hotkeys you know in Edit mode do not exist for texts! The second place is the Button window (Editing context, F9), especially the Font panel.

The menu of the 3D view header has nearly no use, and there is no Specials menu... You have no transform nor mirror tools, and so

on. However, you can apply to texts the same modifiers as to curves.

Editing Text is similar to using a standard text editor but is not as full featured and has some differences:

Exit Edit mode

↔ Tab doesn't insert a tab character in the text, but rather enters and exits Edit mode, as with other object types.

Copy

To copy text to the buffer use CtrlC or the **Copy** button in the tool shelf.

Cut and Copy

To cut and copy text to the buffer use CtrlX or the **Cut** button in the tool shelf.

Paste

To paste text from the buffer use CtrlV or the **Paste** button in the tool shelf.

Delete all text

To completely erase or delete all text use Ctrl← Backspace.

Home/End

↵ Home and → End move the cursor to the beginning and end of a line respectively.

Next/Previous word

To move the cursor on a word's boundary use Ctrl< or Ctrl>.

The text buffer does not communicate with the desktop. It only works from within Blender. To insert text from outside Blender see [Inserting text](#) below.

## Inserting Text

You can insert text in three different ways: from the internal text buffer ([Editing Text](#)), or from a text file.

To load text from a text file, use the Text » Paste File tool. This will bring up a File Browser window for navigating to a valid UTF-8 file. As usual, be careful that the file doesn't have too many characters as interactive response will slow down.

## Special Characters

Mode: Edit mode

Menu: Text » Special Characters

There are a few special characters that are available using the Alt key or the Text menu on the 3D window header.

Here is a summary of these characters:

AltC:	Copyright (©)	AltR:	Registered trademark (®)
AltG:	Degrees (°)	AltX:	Multiply symbol (×)
AltS:	German "ss" (ß)	AltF:	Currency sign (¤)
AltL:	British Pound (£)	AltY:	Japanese Yen (¥)
Alt1:	Superscript 1 (¹)	Alt2:	Superscript 2 (²)
Alt3:	Superscript 3 (³)	Alt:	Circle
Alt?:	Spanish question mark (¿)	Alt!:	Spanish exclamation mark (¡)
Alt<:	Left double quotation mark («)	Alt>:	Right double quotation mark (»)

All the characters on your keyboard should work, including stressed vowels and so on. If you need special characters (such as accented chars, which are not there on a US keyboard) you can produce many of them using a combination of two other characters. To do so, type the main char, press Alt← Backspace, and then press the desired "modifier" to produce the special character. Some examples are given below:

A, Alt← Backspace, ~:	a A, Alt← Backspace, `:	á A, Alt← Backspace, ^:	a
A, Alt← Backspace, O:	a E, Alt← Backspace, ":	ë O, Alt← Backspace, /:	o

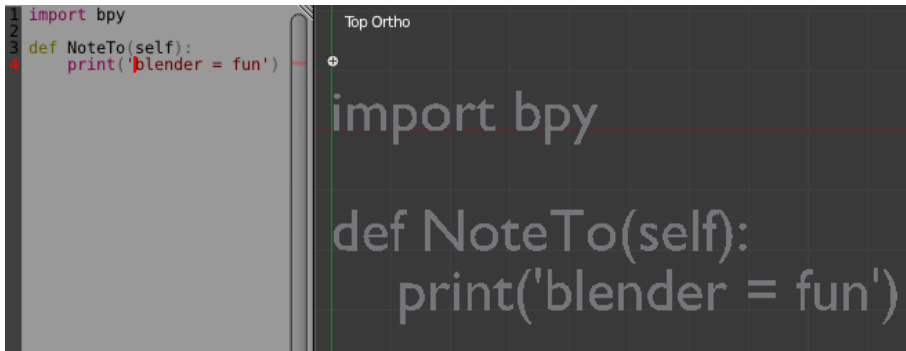
## Convert text to text object



An easy way to get text into Blender is to type it in [The Text Editor](#). It is suggested to do this with a [split window](#) as you will be able to see the 3D view port and text editor at the same time. In the Text Editor select *Text > Create Text Block*. Then begin typing. When finished select *Edit >> Text to 3D Object >> One Object* or *One Object per Line* depending on your needs. It is also possible to load a text file via *Text >> Open Text Block* which can be useful for importing large amounts of text at once.

## 3D Mesh

It is possible to convert a Text Object to a 3D Mesh object. This can be useful so that you may edit the vertices in [Edit Mode](#), but you will lose the ability to edit the text itself. To do this go to [Object Mode](#) and select your Text Object. Press AltC and select *Mesh From Curve/Meta/Surf/Text*. Now you can return to [Edit Mode](#) and manually edit the vertices. They are usually a bit messy, so it may be useful to use a *Limited Dissolve* deletion or *Remesh* Object [Modifier](#) at a low threshold to clean up your mesh.



left normal text, right the made text object.

## Text Selection



Text in Edit mode.

In Edit mode, your text has a white cursor, and as in any text editor, it determines where new chars will be inserted! You move this cursor with the arrow keys (↶/↷/↵) or Page up/Page down and ⌘ Home/⌘ End keys.

Hold ⇧ Shift while using the arrow keys to select a part of the text. You can use it to specify different materials, the normal/bold/italic state, and not much more...

## Formatting Text

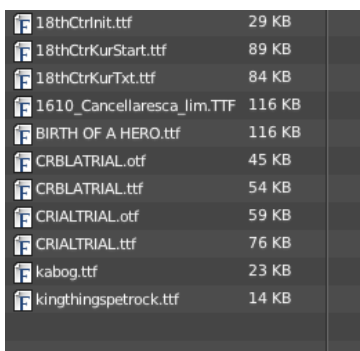
### Fonts

Mode: Edit mode

Panel: Font (Editing context, F9)

The Font panel has several options for changing the look of characters.

### Loading and Changing Fonts



Loading a Type 1 font file.

Blender comes with a *built-in* font by default and is displayed in each of the four font style choosers. The *built-in* font is always present and shows in this list as “Bfont”. The first icon contains a drop-down list displaying currently loaded fonts. Select one for each font style.

To load a different Font, click the **Load** button in the and navigating to a *valid* font. The File Browser window will give all valid fonts a capital F icon, as seen in *Loading a Type 1 font file*.

Unix note

Fonts are typically located under `/usr/lib/fonts`, or some variant like `/usr/lib/X11/fonts`, but not always. They may be in other locations as well, such as `/usr/share/local` or `/usr/local/share`, and possibly related sub-trees.

If you select a font that Blender can't understand, you will get the error "Not a valid font".

Remember the same font will be applied to all chars with same style in a text, but that a separate font is required for each style. For example, you will need to load an *Italics* font in order to make characters or words italic. Once the font is loaded you can apply that font "Style" to the selected characters or the whole object. In all, you would need to load a minimum of four different types of fonts to represent each style (**Normal**, **Italics**, **Bold**, **Bold-Italics**).

It is important to understand that Blender do not bother what font you load for "normal", "bold", etc., style. This is how you can have up to four different fonts in use in the same text – but you have to chose between different styles of a same font, or different fonts. Blender has a number of typographic controls for changing the style and layout of text, found in the Font panel.

## Size and Shear

### Size

Controls the size of the whole text (no way to control each char size independently). Note however that chars with different fonts (different styles, see below) might have different visible size.



shear: 'blender' has a shear value of 1,  
'2.59' a shear value of 0

### Shear

Controls the inclination of the whole text. Even if this seems similar to italics style, *this is not the same thing!*

## Objects as Fonts

You can also "create" your own "font" inside Blender! This is a quite complex process, so let's detail it:

- First, you must create your chars. Each char is an object, *of any type* (mesh, curve, meta...). They all must have a name following the schema: `common prefix` followed by the `char name` (e.g. "ft.a", "ft.b", etc.).
- Then, for the Text object, you must enable the Dupli Verts button (Object context – F7 –, Anim Settings panel).
- Back in Editing context (F9), in the Font panel, fill the Ob Family field with the *common prefix* of your "font" objects.

Now, each time a char in your text matches the *suffix part* of a "font" object's name, this object is duplicated on this char. *The original chars remain visible*. The objects are duplicated so that their center is positioned at the *lower right corner* of the corresponding chars.

## Text on Curve

With the [curve modifier](#) you can let text follow a curve.



Text on curve.

In (*Text on curve*) you can see a text deformed by a curve (a 2D Bézier circle).

to apply the curve modifier, the text object first has to be converted to a mesh, using AltC and click mesh.

### Note

there is also a Text on Curve feature, but the curve modifier offers more options.

## Underline

### Underline

Toggled with the Underline button before typing. Text can also be set to Underlined by selecting it then using the **Bold** button in the Tool Shelf.

### Position

This allows you to shift vertically the position of the underline.

### Thickness

This controls the thickness of the underline.

[File:TextFontSettings.png?](#)

check a character option  
to, for example, type bold  
text

## Character



Bold text.

**Bold**  
Toggled with the Bold button before typing. Text can also be set to Bold by selecting it then using the **Bold** button in the Tool Shelf.

**Italics**  
Toggled with the Italic button before typing. Text can also be set to Italic by selecting it then using the **Italic** button in the Tool Shelf.

**Underline**  
Enables underlining, as controlled by the Underline settings above.

**Small Caps**  
type small capital text.

Blender's Bold and Italic buttons don't work the same way as other applications, as they also serve as placeholders for you to load up certain fonts manually, which get applied when you define corresponding style, see [above](#).

To apply the Bold/Italics/Underline attribute to a set of characters you either turn on Bold/Italics/Underline prior to typing characters, or highlight (select) first and then toggle Bold/Italics/Underline.

## Setting Case

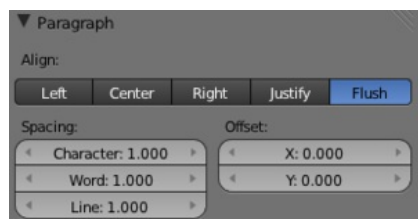
You can change text case by selecting it then clicking the **To Upper** or **To Lower** in the tool shelf.

Enable the Small Caps option to type characters as small caps.

The size of the Small Caps can be changed with the Small Caps Scale setting. Note that the Small Caps Scale is applied the same to all Small Caps formatted characters.

## Paragraph

The Paragraph Panel, has settings for the alignment and spacing of text.



the paragraph tab

## Align

**Left**  
Aligns text to left of frames when using them, else uses the center point of the Text object as the starting point of the text (which grows to the right).

**Center**  
Centers text in the frames when using them, else uses the center point of the Text object as the mid-point of the text (which grows equally to the left and right).

**Right**  
Aligns text to right of frames when using them, else uses the center point of the Text object as the ending point of the text (which grows to the left).

**Justify**  
Only flushes a line when it is **terminated** by a wordwrap (**not** by ↵ Enter), it uses *whitespace* instead of *character spacing* (kerning) to fill lines.

**Flush**

**Always** flushes the line, even when it's still being entered, it uses character spacing (kerning) to fill lines.

Both Justify and Flush only work within frames.

## Spacing

### Character

A factor by which space between each character is scaled, in width

### Word

A factor by which whitespace between words is scaled, in width. You can also control it by selection by pressing Alt+ or Alt+ to decrease/increase spacing by steps of **0.1**.

### Line

A factor by which the vertical space between lines is scaled.

## Offset

### X offset and Y offset

Well, these settings control the X and Y offset of the text, regarding its "normal" positioning. Note that with [frames](#), it applies to all frames' content...

## Shape

Mode: Object or Edit modes

Panel: Curve and Surface (Editing context, F9)

As you can see in the Curve and Surface panel, texts have most of the same options as curves

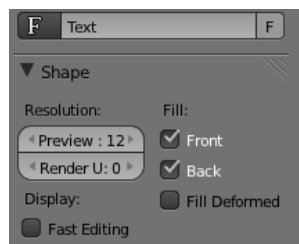
## Resolution

### Preview

the [resolution](#) in the viewport.

### Render

the [resolution](#) on the render.



the shape settings

### Fast Editing

disables curve filling while in edit mode.

## Fill

The fill options control how the text curves are filled in when text is Extruded or Beveled in the Geometry Panel.

### Front

Fills in the front side of the surface.

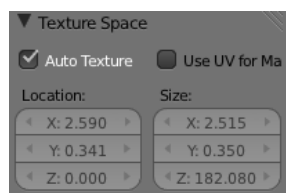
### Back

Fills in the back side of the surface.

### Fill Deformed

Fills the curves after applying shape keys and modifiers.

## Textures



Texture Settings

### Use UV for Mapping

Use UV values as generated texture coordinates.

### Auto Texture Space

Adjusts the active objects texture space automatically when transforming object.

## Geometry

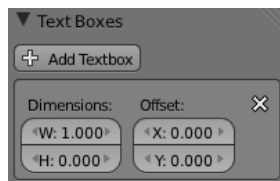
Text objects have all the [curves' extrusion features](#).

Editování textu

## Textové boxy

Mód: Object nebo Edit mód

Panel: Font (Editing kontext, F9)



Textový rámeček.

Textové "Boxy" dávají možnost distribuovat text jako jediný objekt v obdélníkové oblasti a s tím související možnosti libovolného umístění v prostoru.

Text spojitě protéká mezi rámečkem s nejnižším číslem k rámečku s nejvyšším. Pokud je text delší a nevejde se do posledního rámečku, přeteče jej.

Textové rámečky jsou velice podobné konceptu rámečků v aplikacích typu desktop publishing, jako je například nástroj Scribus. Rámeček jako nadřazený prostorový útvar celý text umísťuje do prostoru.

Rámečky jsou ovládány pomocí panelu Text Boxes.

### Velikost rámečku

Počáteční velikost pro nový textový objekt je velikost rámečku **nula**, a to jak pro šířku Width, tak i pro výšku {Literal|Height}. Tyto nulové rozměry způsobují, že počáteční rámeček je neviditelný.

Rámečky se šířkou **0.0** jsou během toku textu ignorovány (žádné přepisování slov) a rámečky s výškou **0.0** protékají vždy (bez možnosti přetečení textu na další rámečky).

A tak aby byl rámeček viditelný, musí být jeho šířka (Width) větší, nežli **0.0**.

Poznámka

Technicky výška není nikdy aktuálně **0.0**, protože písmo (font) sám o sobě má vždy nenulovou výšku.



Šířka rámečku.

(*Frame width*), tedy šířka rámečku je **5.0** a protože je větší nežli nula, je nyní viditelný a vykreslený aktivní barvou tématu jako tečkovaný rámeček. Text přetéká, neb dosáhl konce rámečku.

### Přidání/mazání rámu

Kliknutím na tlačítko **Add Textbox** v panelu Text Boxes se vloží nový textový rámeček právě za aktuální (v pořadí podle toku textu) s nastavenými jeho atributy (pozice a velikost). Ověřte posun tohoto nového rámečku v polích X a Y. Právě úpravou hodnoty X dojde v vytvoření nového sloupce.

Vymazání rámečku je možné kliknutím na tlačítko **Delete**. Text ve vyšších rámečkách přeteče dolů do nižších rámečků.

### Příklad: Tok textu



zalamování

Se dvěma nebo více rámečky je možné je aranžovat text daleko přesněji s dostatečnou volností pohybu. Například přidáním textu a napsáním "Blender je sjůpr - důpr" dostává tento text rámeček. Jen zatím není vidět, protože jeho šíře Width je **0.0**.

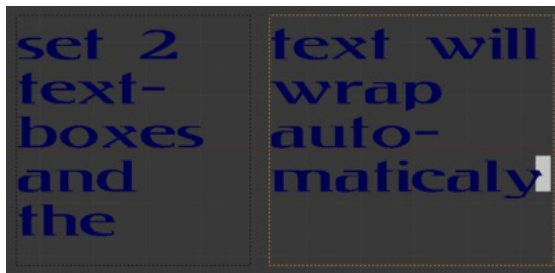
Nastavením šíře na **5.0** se rámeček stává viditelným a text přetéká v souladu s novou šířkou jak je ukázáno v (*Text 2*). Text přetéká konec posledního rámečku, což je způsobeno původní velikostí rámečku.



text protékající z boxu 1 do boxu 2

Pokud přidáme nějaká rámeček a nastavíme jeho šířku a výšku, text proteče to tohoto nového rámků.

### Příklad: Více sloupců



Text 5.

Pro vytvoření dvou sloupců textu pouze vytvořte textový objekt a nastavte původní šířku Width a výšku Height podle požadavků. Pak vložte nový rámeček, a ten bude mít stejnou velikost jako původní rámeček. Sastavte pozici X na hodnotu o něco větší, nebo menší nežli je šířka původního rámků. Viz (*Text 5*).

### Přiřazení materiálu

Mód: Edit mode

Panel: Link and Materials (Editing kontext, F9)

Každý znak může mít jiný index materiálu (Material index), a tak může mít nastaven odlišný materiál.

Přiřazení je možné nastavit před vlastním psaním, nebo po výběrem bloku textu a kliknutím na tlačítko přiřazení **Assign** v panelu materiálů.



Red Green Blue.

Například pro vytvoření obrázku (*Red Green Blue*) nejprve potřebujete vytvořit tři oddělené materiály a tři materiálové indicie. Každé slovo může být přiřazeno Materiálovému indexu výběrem znaků každého slova kliknutím na tlačítko **Assign**. (*Red Green Blue*) je stále jediný samostatný Textový objekt.

Metaobjekty

Mód: Object or Edit modes

Klávesová zkratka: ⇧ ShiftA

Menu: Add » Meta

Meta objekty jsou *implicitní povrchy*, což znamená, že *nejsou explicitně* definovány pomocí vrcholů a hran, nebo kontrolních bodů jako křivky. Existují *procedurálně*. Metaobjekty jsou popsány pomocí matematických formulí, které jsou kalkulovány Blenderem během vykreslování. (kalkulovány za běhu - *calculated on-the-fly*).

Typickou charakteristikou metaobjektů je jejich amorfní tvar podobný tekoucí rtuť, lepidla. Mají raoblený tvar. Navíc v případě dotyku dvou metaobjektů dochází k jejich interakci a slévání jako ve stavu beztíže. Toto může být užitečné například při modelování proudu vody, pokud nechcete jít cestou fluidní simulace kapalin (výpočetně náročné). Pokud se tyto objekty opět vzdálí, zotaví se do svých prvotních tvarů.

Každý z nich je definován základovou [matematickou strukturou](#) a kdykoli je možné mezi nimi přepínat pomocí panelu Active Element panel.

Typicky jsou metaobjekty využívány pro tvorbu speciálních efektů. Je možné například využít skupinu metaobjektů pro vytvoření původních tvarů a po té je konvertovat do objektu jiného typu (sítě) pro následné modelování. Metaobjekty jsou také velice efektivní pro zpracování ray-tracingem.

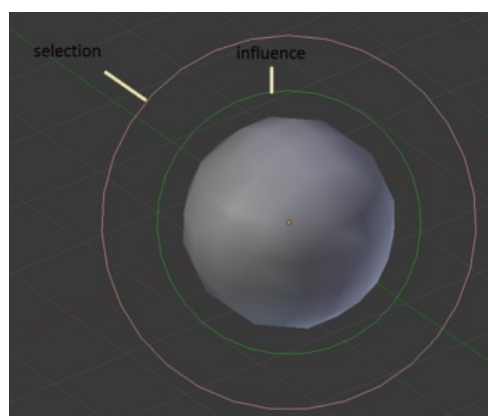
Je vhodné poznamenat, že metaobjekty mají nepatrně odlišné chování v objektovém režimu, což je detailně popsáno [dále](#).

## Primitiva

V menu Add » Meta je k dispozici těchto pět základních "primitiv" (nebo konfigurací):

- Meta Ball přidá "míč" s bodem popisujícím strukturu.
- Meta Tube přidá "trubicu" se segmentem úsečky popisujícím strukturu.
- Meta Plane přidá "rovinu" s planární strukturou.
- Meta Elipsoid přidá "elipsoid" s popisnou strukturou.
- Meta Cube přidá "kostku" s objemovým popisem struktury.

## Vizualizace



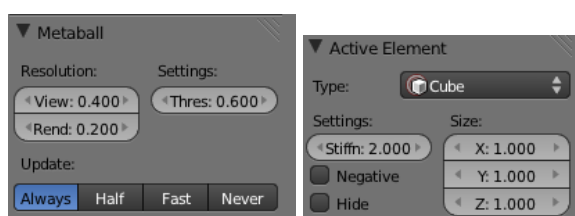
Příklad Meta Ball. (selection - výběr, influence - vliv)

V objektovém režimu je zobrazena kalkulovaná síť spolu s čeným prstencem "selection ring" (pokud je vybraný, je růžový). Více o chování metaobjektů v objektovém režimu viz [níže](#).

V editačním režimu (*Meta Ball příklad*) je vykreslen jako síť (stínovaná, nebo jako černá drátová, ale samozřejmě bez jakékoli hrany a vrcholu) se dvěma barevnými kruhy: červený pro výběr (vybraný je růžový) a zelený pro přímé řízení pružnosti (viz [níže](#) – světle zelený, pokud je aktivní). Je vhodné poznamenat, že pokud nedojde ke škálování škálování (S), je vyznačená kružnice ekvivalentní červené kružnici.

## Volby pro Meta Ball

Šechny metaobjekty na scéně interagují navzájem. Nastavení v sekci MetaBall působí na všechny metaobjekty. V editačním režimu je zobrazen panel Active Element umožňující editaci jednotlivých meta-elementů.



Celkové, globální meta vlastnosti.

Individuální meta vlastnosti.

## Rozlišení

The Resolution controls the resolution of the resultant mesh as generated by the Meta object.

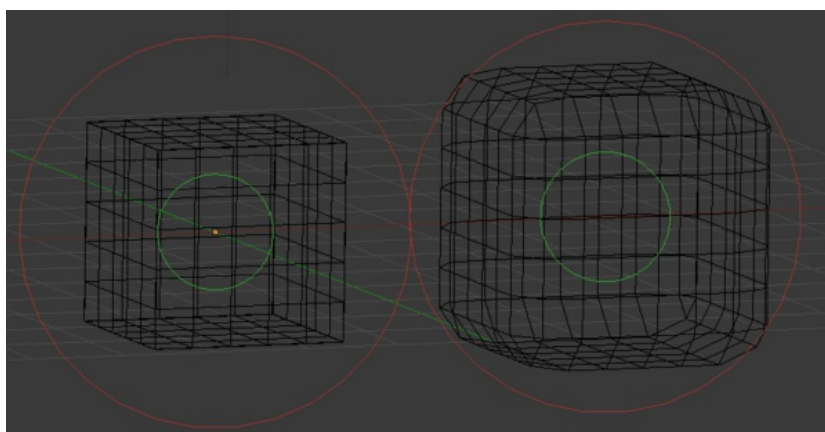
### View

The 3D View resolution of the generated mesh. The range is from **0.05** (finest) to **1.0** (coarsest).

### Render

The rendered resolution of the generated mesh. The range is from **0.05** (finest) to **1.0** (coarsest).

One way to see the underlying mathematical structure is to lower the Resolution, increase the Threshold and set the Stiffness (see below) a fraction above the Threshold. (*Underlying structure*) is a (*Meta cube*) with the above mentioned configuration applied as follows: Resolution of **0.410**, Threshold of **5.0** and Stiffness a fraction above at **5.01**.



Left: Underlying structure, Right: the shape.

You can clearly see the underlying cubic structure that gives the meta cube its shape.

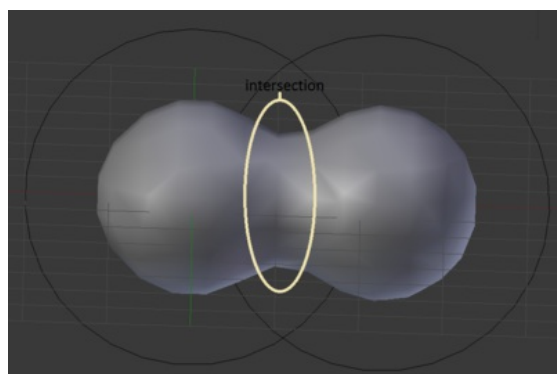
## Threshold (Influence)

Mode: Object or Edit modes

Panel: MetaBall (Editing context, F9)

Threshold defines how much a meta's surface "influences" other metas. It controls the *field level* at which the surface is computed. The setting is global to a [group](#) of Meta objects. As the threshold increases, the influence that each meta has on one another increases.

There are two types of influence: **positive** or **negative**. The type can be toggled on the Active Element panel while in Edit mode, using the Negative button. You could think of **positive** as attraction and **negative** as repulsion of meshes. A negative meta will push away or repel the meshes of positive Meta objects.



Positive.

A *positive* influence is defined as an attraction meaning the meshes will stretch towards each other as the *rings of influence* intersect. (*Positive*) shows two meta balls' *ring of influence* intersecting with a *positive* influence.

Notice how the meshes have pulled towards one another. The area circled in white shows the green *influence* rings intersecting.

## Update

While transforming metas (grab/move, scale, etc.), you have four "modes" of visualization, located in the Update buttons group of the MetaBall panel:

- Always – fully draw the meta during transformations.
- Half Res – During transformations, draw the meta at half its Wiresize resolution.
- Fast – Do not show meta mesh during transformations.
- Never – Never show meta mesh (not very recommended option, as the meta is only visible at render time!).

This should help you if you experiment difficulties (metas are quite compute-intensive...), but with modern computers, this shouldn't happen, unless you use many metas, or very high resolutions...

## Meta Structure

### Technical Details

A more formal definition of a meta object can be given as a *directing structure* which can be seen as the source of a static field. The field can be either positive or negative and hence the field generated by neighboring directing structures can attract or repel.

The implicit surface is defined as the surface where the 3D field generated by all the directing structures assume a given value. For example a meta ball, whose directing structure is a point, generates an isotropic (i.e. identical in all directions) field around it and the surfaces at constant field value are spheres centered at the directing point.

Meta objects are nothing more than mathematical formulae that perform logical operations on one another (AND, OR), and that can be added and subtracted from each other. This method is also called **Constructive Solid Geometry** (CSG). Because of its mathematical nature, CSG uses little memory, but requires lots of processing power to compute.

### Underlying Structure

Mode: Edit mode

Panel: MetaBall tools (Editing context, F9), Transform Properties

Blender has five types of metas, each determined by its underlying (or directing) structure. In Edit mode, you can change this structure, either using the relevant buttons in the MetaBall tools panel, or the drop-down list in the Transform Properties panel (N). Depending on the structure, you might have additional parameters, located in both Transform Properties and MetaBall tools panels.

Ball (point, zero-dimensional structure)

This is the simplest meta, without any additional setting. As it is just a point, it generates an isotropic field, yielding a spherical surface (this is why it is called Meta Ball or Ball in Blender).

Tube (straight line, uni-dimensional structure)

This is a meta which surface is generated by the field produced by a straight line of a given length. This gives a cylindrical surface, with rounded closed ends. It has one additional parameter:

- dx: The length of the line (and hence of the tube – defaults to **1.0**).

Plane (rectangular plane, bi-dimensional structure)

This is a meta which surface is generated by the field produced by a rectangular plane. This gives a parallelepipedal surface, with a fixed thickness, and rounded borders. It has two additional parameters:

- dx: The length of the rectangle (defaults to **1.0**).
- dy: The width of the rectangle (defaults to **1.0**).

Note that by default, the plane is a square.

Ellipsoid (ellipsoidal volume, tri-dimensional structure)

This is a meta which surface is generated by the field produced by an ellipsoidal volume. This gives an ellipsoidal surface. It has three additional parameters:

- dx: The length of the ellipsoid (defaults to **1.0**).
- dy: The width of the ellipsoid (defaults to **1.0**).
- dz: The height of the ellipsoid (defaults to **1.0**).

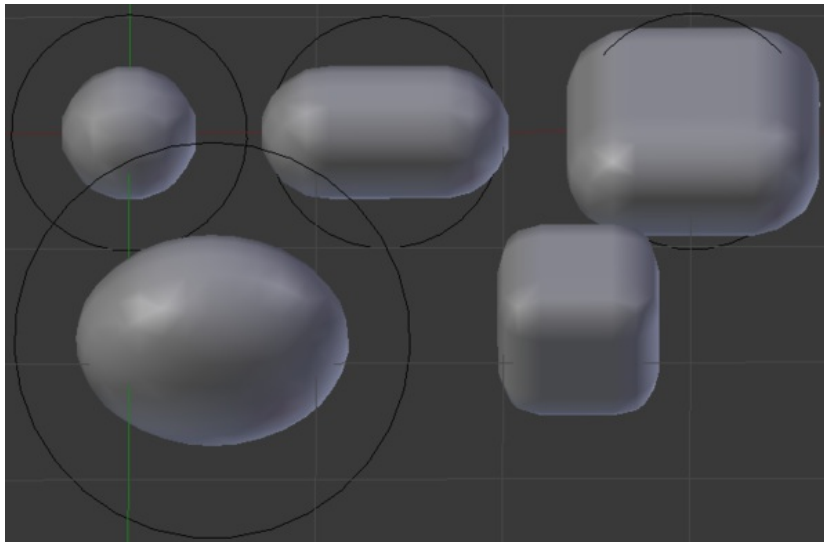
Note that by default, the volume is a sphere, producing a spherical meta, as the Ball option...

Cube (parallelepipedal volume, tri-dimensional structure)

This is a meta which surface is generated by the field produced by a parallelepipedal volume. This gives a parallelepipedal surface, with rounded edges. As you might have guessed, it has three additional parameters:

- dx: The length of the parallelepiped (defaults to **1.0**).
- dy: The width of the parallelepiped (defaults to **1.0**).
- dz: The height of the parallelepiped (defaults to **1.0**).

Note that by default, the volume is a cube.



the 5 meta primitives.

## Prázdné objekty (Empties)

"Prázdný" objekt je nulový objekt. V textu dále někdy uveden pojem "Prázdný". Prázdný objekt neobsahuje žádná reálná geometrická data týkající se tvaru, ale může být využit mnoha způsoby.

## Nastavení

### Plain Axes (osy)

Vykreslí šest polopřímek směřujících podél os +X,-X,+Y,-Y,+Z a -Z.

### Arrows (šipky)

Vykreslí šipky ukazující kladným směrem os X,Y, a Z. Každou s popisem.

### Single Arrow (jedna šipka)

Vykreslí jednu šipku směřující podél osy +Z kladným směrem.

### Circle (kruh)

Vykreslí kružnici v rovině XZ.

### Cube (krychle)

Vykreslí rám (hrany) krychle.

### Sphere (koule)

Vykreslí kouli pomocí tří kružnic.

### Cone (hranol)

Vykreslí hranol směřující podél kladné osy Y.

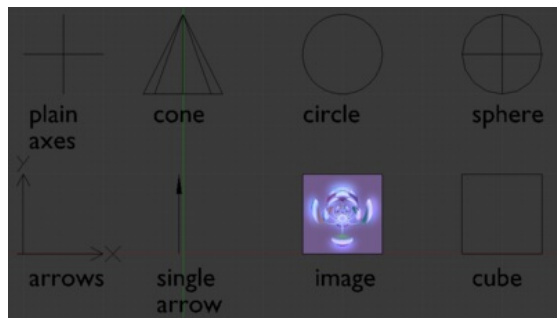
### Image (obrázek)

Prázdné objekty mohou také zobrazovat obrázky. Tato vlastnost může být využita pro vytváření referencí, záložek a popisů v modelu namísto použití obrázků v pozadí. Obrázek se zobrazí v souladu s modelovaným prostředím 3D. Nastavení je stejné jako v [Background Image Settings](#)

*Poznámka: Pokud jsou použity alfa-obrázky (s průsvitností), je zde omezení dané pořadí vykreslování objektů, kde průsvitná alfa nebude vykreslena nahoře při výběru různých objektů.*

### Size (velikost)

Řídí lokální velikost prázdného objektu. Nemění velikost ale jednoduše změní tvar.



Osm různých typů vykreslení prázdného objektu

## Použití a funkce

Prázdné objekty lze s úspěchem využít jako transformační prvek, ačkoli nejsou ve výsledku sami renderovány. Jsou důležité a velice užitečné v mnohých oblastech. Některé příklady toto demonstrují:

### Rodičovský objekt pro skupinu dalších objektů

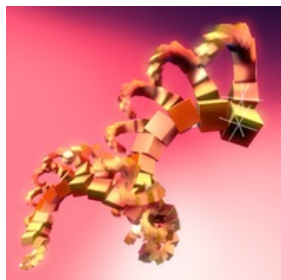
- Prázdný může být rodičem libovolného počtu dalších objektů. To dává uživateli možnost jednoduše je řídit.

### Omezovací cíl (Target for constraints)

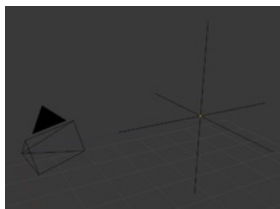
- Prázdný je možné také použít jako cílový bod pro normály, nebo pro omezení kostí..
- To dává uživateli mnohem více možností řízení, například nastavení kamery pro sledování určitého bodu pomocí její funkce **Track to**.

### Posun pole (Array offset)

- Prázdný může být zapojeno jako specifikátor posuvu pro modifikátor pole, a tak zajistit funkce velice složitých deformací pouhým pohybem jednoho objektu



Příklad Prázdná pro řízení pole objektů



Příklad Prázdna použitého  
pro funkci sledování

Jiné použití.

- Umísťovací značky
- Kontrola pohybu
- Vzdálenost DOF (zaostření objektivu)
- Referenční obrázky

## Group Instances

Skupiny jsou detailně popsány v následujícím odkazu: [Seskupování a rodičovské vazby](#).

Pokud ve scéně existuje skupina, pak může být vytvořena její instance, a to pomocí menu Add v položce add » group instance » group.

Tyto dvě skupiny jsou kontrolovány/řízeny/vlastněny dodatečným prázdným objektem, a tak nejdou editovatelné (tj. není pro ně povolen editovací režim) – in fact, they behave a bit as if all object copies of the group where children of the empty). So you have all the editing options of objects (you can move/scale/rotate them, etc.).

## Groups and Metas

There seems to be a bug with Meta objects in group proxies: their meshes are invisible, only the circles are drawn...

## Vizualizace

"Zástupce" skupin jsou vykresleny černě (saozřejmě pokud je nastaven režim Solid-pevný nebo Shaded-stínovaný ).Tyto výběry "skutečných"členů skupin jsou The selection of "real" members of the group is odrazem jejich "zástupců". Pokud je vybrán zástupce, prázdný objekt se přepne do růžové barvy a ostatní části do fialové.

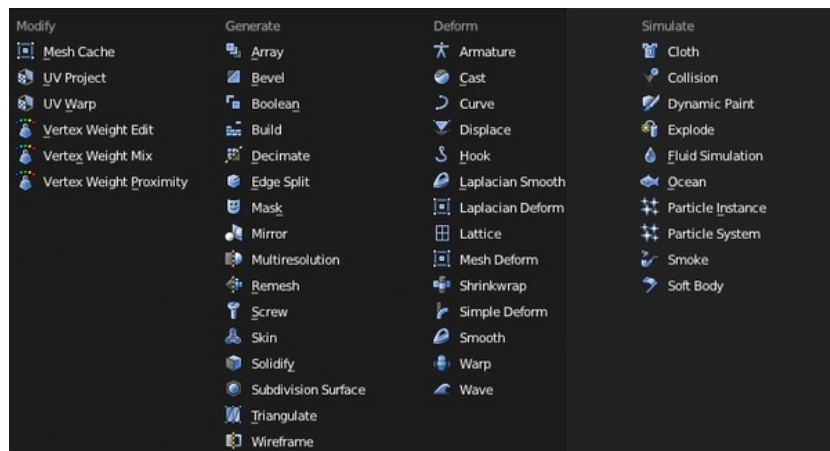
Tato možnost "zástupců skupin"je stejná jako v popisu [vizualizace prázdných objektů](#).

## Modifikátory

Mód: libovolný

Panel: Modifiers

Modifikátory jsou automatické operace, které nedestruktivním způsobem upravují objekt. S modifikátory je možné zajistit vytvoření mnoha efektů které by bylo velice složité vytvořit manuální úpravou objektu (jako rozdělení povrchu), nebo bez vlivu na celkovou topologii objektu. Modifikátory působí na změnu toho, jak bude objekt zobrazen a renderován a nikoli na jeho samotnou geometrii. V jednotlivému objektu je možné přidat více modifikátorů a vytvořit tak zásobník modifikátorů ([Svazek modifikátorů](#)) a je možné aplikovat (Apply) modifikátor až tehdy, kdy je vhodné způsobit změny trvalé.



Menu čtyř skupin modifikátorů.

## Úprava (Modify)

Skupina Modify jsou nástroje velice podobné deformační skupině, ale přímo nepůsobí na tvar objektu. Mají vliv na data objektu jako je skupina vrcholů sítě (vertex groups)...

### [Mesh Cache](#)

Způsobí animaci síťových dat z externího souboru.

### [UV Project](#)

Projektuje souřadnice UV na upravovanou síť.

### [UV Warp](#)

Dynamicky jsou editovány souřadnice UV na dané síti.

### [Vertex Weight](#)

Umožňuje úpravy vrcholů různými způsoby.

## Vytváření (Generate)

Skupina modifikátorů Generate patří mezi konstrukční nástroje, které mohou automaticky přidávat a vytvářet nové objekty.

### [Array](#)

Vytváření pole (zástupu) opakujících se tvarů.

### [Bevel \(Zaoblení\)](#)

Vytvoří zaoblení na vybranou modelovou síť.

### [Boolean](#)

Kombinuje/odečítá/protíná síť s jinou.

### [Build \(vytvoření\)](#)

Sestaví během animace síť krok po kroku.

### [Decimate \(decimovat\)](#)

Redukuje počet polygonů sítě.

### [Edge Split \(oddělení hran\)](#)

Přidá tvarové hrany do sítě.

### [Mask \(maskování\)](#)

Umožňuje skrýt některé části sítě.

### [Mirror \(zrcadlení\)](#)

Zrcadí objekt podle jedné z os a výsledkem je symetrická sít.

### [Multiresolution](#)

Mění požadovanou síť objektu podle různých úrovní rozlišení.

### [Remesh](#)

Může opravit husté trojúhelníkové sítě a pomoci jemného doladění prahu citlivosti..

### [Screw](#)

Vytváří geometrii helix-pattern z jednoduchého profilu. Je podobné kroučicímu Screw nástroji při použití na síť.

### [Skin](#)

Automatická tvorba topologie.

### [Solidify](#)

Dává hloubku síťovým plochám.

### [Subdivision Surface](#)

Rozzdělí síť pomocí metody Catmull-Clark nebo jednoduchého algoritmu.

### [Triangulate](#)

Triangulace. Předevení všech povrchů na trojúhelníky.

## Deformace (Deform)

The Deform group of modifiers only change the shape of an object, and are available for meshes, and often texts, curves, surfaces and/or lattices.

### [Armature](#)

Use bones to deform and animate your object.

### [Cast](#)

Shift the shape of a mesh, surface or lattice to a sphere, cylinder or cuboid.

### [Curve](#)

Bend your object using a curve as guide.

### [Displace](#)

Deform your object using a texture.

### [Hook](#)

Add a hook to your vertice(s) (or control point(s)) to manipulate them from the outside.

### [Laplacian Smooth](#)

Allows you to reduce noise on a mesh's surface with minimal changes to its shape.

### [Lattice](#)

Use a Lattice object to deform your object.

### [Mesh Deform](#)

Allows you to deform your object by modifying the shape of another mesh, used as a "Mesh Deform Cage" (like when using a lattice).

### [Shrinkwrap](#)

Allows you to shrink/wrap your object to/around the surface of a target mesh object.

### [Simple Deform](#)

Applies some advanced deformations to your object.

### [Smooth](#)

Smooth the geometry of a mesh. Similar to the Smooth tool in the mesh editing context.

### [Warp](#)

Warp a mesh by specifying two points the mesh stretches between.

### [Wave](#)

Deform your object to form (animated) waves.

## Simulace

The Simulate group of modifiers activate simulations. In most cases, these modifiers are automatically added to the modifiers stack whenever a Particle System or Physics simulation is enabled, and their only role is to define the place in the modifiers stack used as base data by the tool they represent. Generally, the attributes of these modifiers are accessible in separate panels.

### [Cloth](#)

Simulates the properties of a piece of cloth. It is inserted in the modifier stack when you designate a mesh as Cloth.

### [Collision](#)

Simulates a collision between objects.

### [Explode](#)

Blows up your mesh using a particle system.

### [Fluid](#)

The object is part of a fluid simulation... The modifier added when you designate a mesh as Fluid.

### [Particle Instance](#)

Makes an object act similar to a particle but using the mesh shape instead.

### [Particle System](#)

Represents a particle system in the stack, so it is inserted when you add a particle system to the object.

### [Smoke](#)

Simulates realistic smoke.

### [Soft Body](#)

The object is soft, elastic... Modifier added when you designate a mesh as Softbody.

### [Dynamic Paint](#)

Makes an object or a particle system paint a material onto another object.

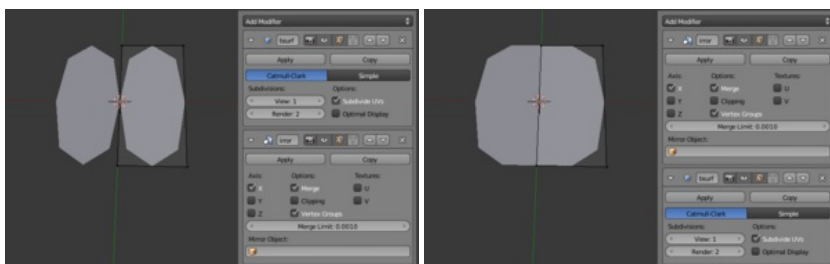
### [Ocean](#)

Quickly creates a realistic, animated ocean.

## Modifikátor

Modifikátor je definovaný jako použití "procesu" neb "algoritmu" na objekt. Mohou být použity interaktivně a nedestruktivně podle přání uživatele. Tato vlastnost je někdy popisována jako zásobník, nebo "svazek modifikátorů" (modifier stack) a nachází se v mnoha jiných aplikacích typu 3D. (fakticky svazek je souhrn několika modifikátorů, které jsou svázány pořadím)

Modifiers are added from the Modifiers menu. Some tools and scripts, for example Decimate and Solidify, have been migrated from its previous location and changed into a modifier. In a modifier stack the order in which modifiers are applied has an effect on the result. Fortunately modifiers can be rearranged easily by clicking the convenient up and down arrow icons. For example, (*Stack ordering*) shows [SubSurf](#) and [Mirror](#) modifiers that have switched places.

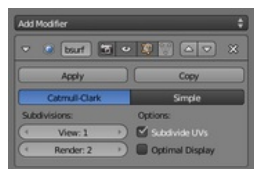


Stack ordering

In the first example, the Mirror modifier is the last item in the stack. The result looks like two surfaces. In the other example the mirror modifier is the first item in the stack and the result is a single merged surface.

You can see that the results look different from the previous. This means that the stack order is very important in defining the end results.

## Rozhraní (Interface)



Panel Layout (Subsurf as an example)

Each modifier has been brought in from a different part of Blender, so each has its own unique settings and special considerations. However, each modifier's interface has the same basic components, see (*Panel Layout (Subsurf as an example)*).

At the top is the panel header. The icons each represent different settings for the modifier (left to right):

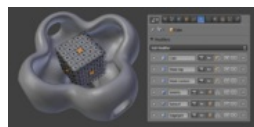
1. Arrow — Collapse modifier to show only the header.
2. Modifier icon and a box for the name of this modifier — default being the name of the modifier itself. It is unique amongst other modifiers of the same type.
3. Camera — Display modifier effect during render time.
4. Eye — Display modifier effect in the 3D view.
5. Box — Shows modifier effect in Edit mode. This button may not be available depending on the type of modifier.
6. Triangle — Applies modifier to editing cage in Edit mode. This icon materializes the Cage mode.
7. Up arrow — Moves modifier up in the stack.
8. Down arrow — Moves modifier down in the stack.
9. Cross — Removes the modifier from the stack.

Below the header are two buttons:

1. Apply — Makes the modifier real.
2. Copy — Creates a copy of the modifier at the base of the stack.

And below these buttons is a sub panel with settings for individual modifiers.

## Svazek (Stack)



In this example a simple subdivided cube has been transformed into a rather complex object using a stack of modifiers.

([blend](#))


To add a modifier you add it to the *stack*. Once added (always at the bottom of the stack), they can be rearranged to your liking.

Some modifiers can only be applied to certain object types. This is indicated by the panel filtering the Add Modifier button on the

Modifiers panel. Only modifiers that can be applied are shown in this listbox button. Mesh objects can have all available modifiers added, while, for example, Lattice objects type objects can only have a few.

Modifikátor zaoblování (bevel)

Mód: Objektový režim

Panel: Properties Window -> Context Button Modifiers 

Modifikátor Bevel způsobuje zaoblení hran tělesa podél jeho hran. Kontrolními parametry je možné volit různé stupně a styly zaoblení. Obvykle jako jiné modifikátory je možné jej aplikovat jednorázově (tzn. pomocí tlačítka Apply), kdy vzniknou nové plochy a hrany obroubeného tělesa. Obvykle je během modelování užitečnější nedestruktivní volba ponecháním omodifikátoru s možností budoucích změn jeho parametrů.

Dá se říci, že modifikátor Bevel je nedestruktivní alternativou k operacím obrábění [Bevel Operation](#) v editačním režimu sítě.

Další obrázky ukazují zaoblený a nezaoblený čtverec, jehož hrany svírají úhel 90° (kolmice). Modifikátor může zpracovat nejen 3D objekty, ale i rovinné 2D objekty.



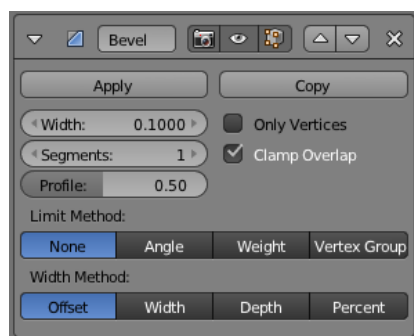
Základní oblení krychle.



Nezaoblený  
čtverec.

Zaoblený  
čtverec.

## Volby



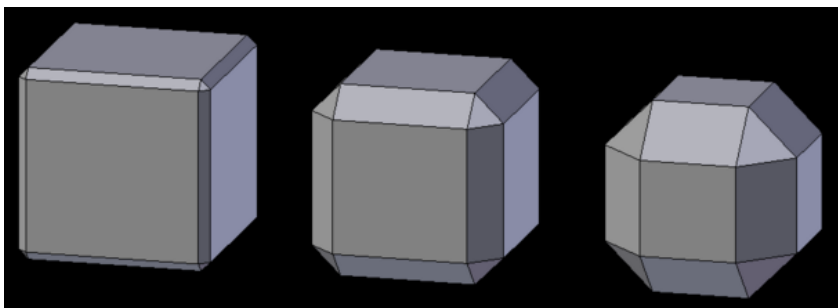
Bevel modifier panel.

## Šířka

The Width numeric field controls the width/amount of the bevel applied to the base mesh. It can range from **0.0** (no bevel applied) to **1.0** (Blender Units). This value is in fact the “backing up” of the two new edges relatively to the original (beveled) one, along the two concerned faces.

### Note

When using Metric or Imperial units the Width value has a unit. E.g. when 1 Blenderunit is 1m a useful value is between 0cm and 100cm. When it seems that decreasing the Width has no effect anymore check if the unit changed to m instead of cm.



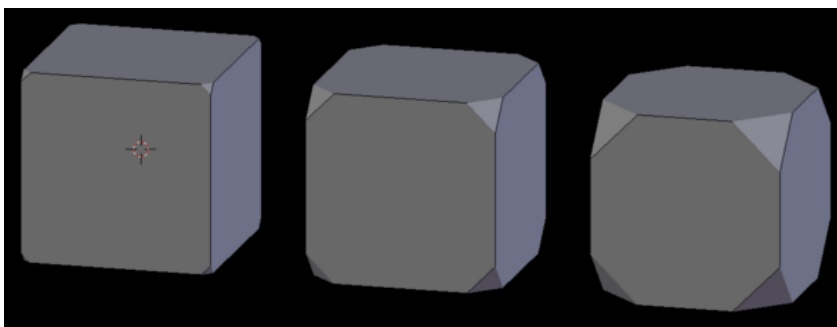
Tři krychle se šířkou (Width) zaoblením **0.1**, **0.3** a **0.5**.

## Segmenty

Nastavuje počet oblých segmentů.

## Pouze vrcholy (Only Vertices)

Tlačítko Only Vertices přepíná do režimu oblení vrcholů (rohů). Po zapnutí jsou obloubeny pouze vrcholy a nikoli hrany..



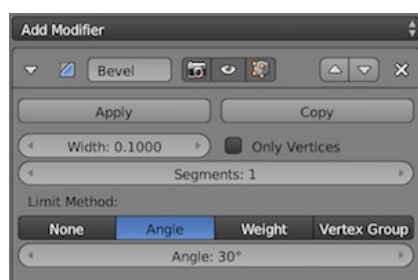
Tři krychle se šířkou (Width) zaoblením **0.1**, **0.3** a **0.5** se zapnutou volbou pouze vrcholy (Only Vertices).

### Metody omezení (Limit Method)

This section of the Bevel modifier is used to control where and when a bevel is applied to the underlying mesh. The first row of three buttons (mutually exclusive) controls the algorithm used, and might add some extra options.

#### None

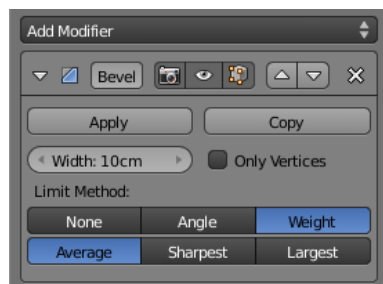
This button will apply the Bevel modifier to the whole underlying mesh, without any way to prevent the bevel on some edges/vertices.



Bevel modifier with the Angle limit enabled.

#### úhel (Angle)

This button will only bevel edges where faces make sharp angles. When selected, it displays the Angle numeric field, used to set the angle above which an edge will be beveled (it is in fact the complementary angle, i.e.  $180^\circ - (\text{angle between faces})$ ). When the angle between meeting faces is less than the angle in the slider box, a bevel on those specific edges will not be applied. Similarly, when the angle between two edges is less than this limit, the vertex is not beveled.



Bevel modifier with Weight button active.

#### Weight

Use bevel weights to determine how much bevel is applied; apply them separately in vert/edge select mode. See [Here](#) about adjusting bevel weights. The three options specify what edge weight to use for weighting a vertex.

##### Average

Uses the average bevel weight at the vertex

##### Sharpest

Uses the smallest bevel weight at the vertex

##### Largest

Uses the largest bevel weight at the vertex

#### Vertex Group

Use a vertex group to determine which parts of the mesh get beveled.

Modifikátor zjednodušení (Decimate)

Mód: Object režim

Panel: Modifiers

## Popis

Modifikátor Decimate (zdecimovat) umožňuje snížit počet vrcholů/stěn s minimálními změnami tvaru tělesa. Toto však takto nefunguje u modelů, které byly vytvářeny pečlivě a ekonomicky, ale naopak má modifikátor význam u těch sítí, které vznikly nějakou operací, jsou složité, komplikované a nepřehledné, nebo tam, kde je velké množství vrcholů a hran v modelu zbytečných.

The Decimate modifier is a quick and easy way of reducing the polygon count of a mesh non-destructively. This modifier demonstrates the advantages of a mesh modifier system because it shows how an operation which is normally permanent and destroys original mesh data, can be done interactively and safely using a modifier.

Unlike the majority of existing modifiers, the Decimate modifier does not allow you to visualize your changes in Edit mode.

Decimate only handles triangles, so each quadrilateral face is implicitly split into two triangles for decimation.

## Volby



Zjednodušující modifikátor decimate

### Ratio

The poměr (ratio) of faces to keep after decimation, from **0.0** (0%, all faces have been completely removed) to **1.0** (100%, mesh is completely intact, except quads have been triangulated).

As the percentage drops from **1.0** to **0.0**, the mesh becomes more and more decimated until it no longer visually looks like the original mesh.

### Face Count (display only)

This field shows the number of remaining faces as a result of applying the Decimate modifier.

## Příklady

### Jednoduchá rovina

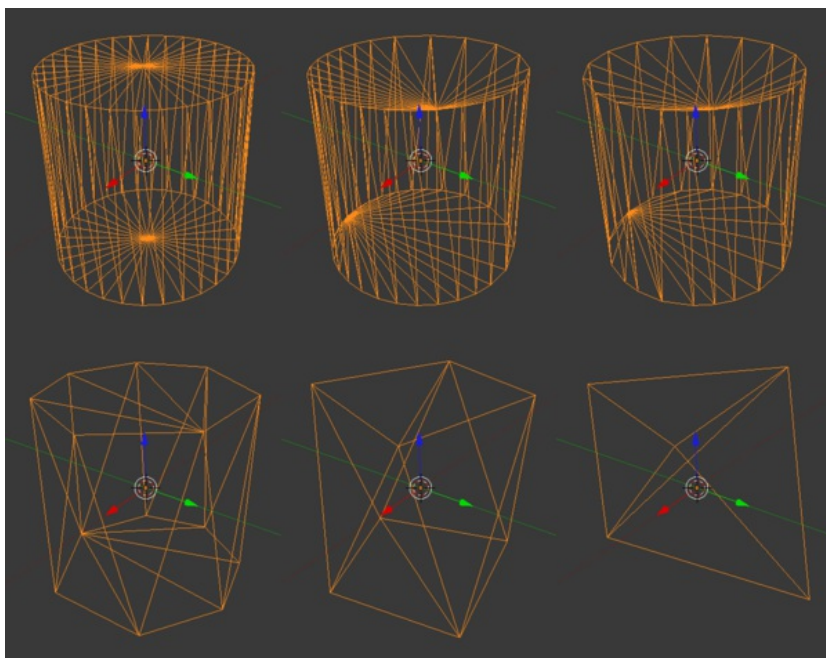
A simple example is a plane, and a 4x4 undeformed Grid object. Both render exactly the same, but the plane has one face and four vertices, while the grid has nine faces and sixteen vertices, hence lots of unneeded vertices and faces. The Decimate modifier allows you to eliminate these unneeded faces.

### Zjednodušený válec

We take a simple example of decimating a cylinder with the default of **32** segments. This will generate a cylinder with **96** faces. When the Decimate modifier is applied, the face count goes up! This is because the modifier converts all quadrangles (*quads*) into triangles (*tris*) which always increases the face count. Each quad decomposes into two triangles.

The main purpose of the Decimate modifier is to reduce mesh resources through a reduction of vertices and faces, but at the same time maintain the original shape of the object.

In the following picture, the percentage dropped in each successive image, from **100%** to **5%** (a ratio of **0.05**). Notice that the face count has gone from **128** to **88** at a ratio of **0.6** (**60%**) and yet the cylinder continues to look very much like a cylinder, and we discarded **40** unneeded faces.



1.0 (100%). Faces: 128; 0.8 (80%). Faces: 102; 0.6 (60%). Faces: 88  
0.2 (20%). Faces: 24; 0.1 (10%). Faces: 12; 0.05 (5%). Faces: 6

As you can see, when the ratio reaches **0.1**, the cylinder looks more like a cube. And when it reaches **0.05**, it doesn't even look like a cube!

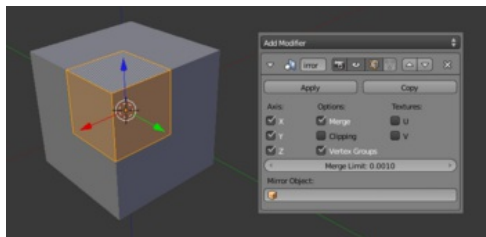
Once you have reached the face count and appearance you were looking for, you can Apply the modifier. If you want to convert many of the tris back to quads to reduce mesh resources further, you can switch to Edit mode, select all vertices (A), and hit AltJ.

Modifikátor zrcadlení

Mód: libovolný režim

Panel: Modifiers

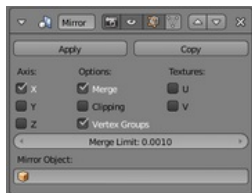
## Popis



Roh krychle zrcadlený podél všech tří os vytvoří..  
nuže... vytvoří krychli.

The Mirror modifier automatically mirrors a mesh along its **local** X, Y and/or Z axes, which pass through the object's center (the mirror plane is then defined by the two other axes). It can also use another object as mirror center, then use that object's local axes instead of its own. It can weld vertices together on the mirror plane within a specified *tolerance* distance. Vertices from the original object can be prevented from moving across or through the mirror plane. And last but not least, it can also mirror vertex groups and UV coordinates.

## Volby



Mirror modifier

### Axis

The axis (X, Y, or Z) along which to mirror (i.e. the axis perpendicular to the mirror plane of symmetry). To understand how the axis applies to the mirror direction, if you were to mirror on the X axis, the X plus values of the original mesh would become X minus values on the mirrored instance.

You can select more than one of these axes – you'll then get more mirror instances, so that all planes of symmetry selected are “fully processed” (i.e. with one axis you get a single mirror, with two axes four mirrors, and with all three axes eight mirrors).

### Options

#### Merge

Merges vertices at the mirror plane. See Merge Limit below.

#### Clipping

Prevents vertices from crossing through the mirror plane(s). Note that this is only valid in Edit mode (i.e. when using object transformations, translations, scaling, et cetera, in Object mode, vertices will happily cross these borders.)

If Clipping is selected but vertices are outside of the Merge Limit the vertices will not merge. As soon as the vertices are within Merge Limit they are clipped together and cannot be moved beyond the mirror plane. If several vertices are selected and are at different distances from the mirror plane, they will one by one be clipped at the mirror plane.

Once you have confirmed clipped vertices with LMB you must, if you want to break the clipping, un-select Clipping to be able to move vertices away from the mirror.

#### Vertex Groups

When this button is enabled, the Mirror modifier will try to mirror existing vertex groups. A very nice feature, but that has quite specific prerequisites.

- First, the vertex groups you want to mirror must be named following the usual left/right pattern (i.e. suffixed by something like “.R”, “.right”, “.L”, et cetera).
- Next, you must have the “mirrored” groups already existing (i.e. same names suffixed by the “other side”) *and completely empty* (no vertex assigned to it), else it won't work.

Usually, the mirrored copies of the vertices of a group remain in this group. Once this option is activated, all groups following the rules described above will only be valid on the original object – the mirrored copy will put these same vertices into the “mirror” group. Very handy with armatures, for example: you just model half of your object, carefully rig it with half of your armature, and just let the Mirror modifier build the other half. Just be sure to put your Armature modifier(s) after the Mirror one.

A final word about multi-axes mirror: in these cases, the “direct”, “first level” copies get the mirrored groups, the copies of copies (“second level”) get the original groups, et cetera.

### Textures

The U and V options allows you to mirror, respectively, the U and V texture coordinates. The values are “mirrored” around the **0.5** value, i.e. if you have a vertex with UV coordinates of (**0.3**, **0.85**), its mirror copy will have UV coordinates of (**0.7**, **0.15**) with both buttons enabled.

### Merge Limit

The maximal distance between vertices and mirror plane for the welding between original and mirrored vertices to take place.

The vertices then will snap together, allowing linking the original mesh to its mirrored copy.

### Mirror Object

The name of another object (usually an empty), to be used as the reference for the mirror process: its center and axes will drive the plane(s) of symmetry. You can of course animate its position/rotation (lpo curves or others), to animate the mirror effect.

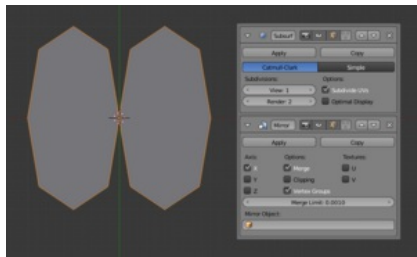
## Hints

Many modeling tasks involve creating objects that are symmetrical. However, there used to be no quick way to model both halves of an object without using one of the workarounds that have been discovered by clever Blender artists over the years. A common technique is to model one half of an object and use AltD to create a linked duplicate which can then be mirrored on one axis to produce a perfect mirror-image copy, which updates in real time as you edit.

The Mirror modifier offers another, simpler way to do this. Once your modeling is completed you can either click Apply to make a real version of your mesh or leave it as is for future editing.

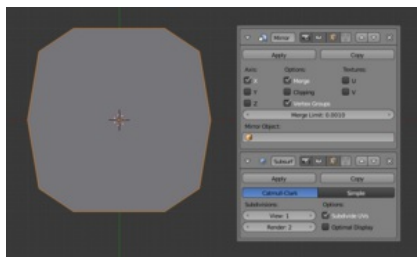
### Použití modifikátoru zrcadlení spolu s modifikátorem podrozdělení povrchu (Subdivision Surface)

When using the Mirror modifier along with the Subsurf modifier, the order in which the modifiers are placed is important.



Subsurf modifier before Mirror modifier

This shows the Subsurf modifier placed before the Mirror one; as you can see the effect of this is that the mesh splits down the center line of the mirror effect.



Mirror modifier before Subsurf modifier

This shows the Mirror modifier placed before the Subsurf modifier. In this order you will get the the center line of the mesh snapped to the center line, which in most cases would be the desired effect.

### Zarovnávání se zrcadlem

To apply a Mirror modifier, it is common to have to move the object's center onto the edge or face that is to be the axis for mirroring. This can be tricky when attempted visually. A good technique to achieve an exact position is to determine the edge against which you wish to mirror. Select two vertices on that edge. Then use **⇧ ShiftS** followed by **C**ursor to **S**election (**C**). This will center the 3D cursor exactly on the edge midway between the two vertices. Finally, press **CtrlAlt⇧ ShiftC** for the Set Origin popup, then select **O**rigin to **3D C**ursor (**T**). This will move the object's center to where the 3D cursor is located, and the mirroring will be exact.

Alternativou je využití prázdného Empty objektu jako objekt zrcadlení (Mirror Object), který je možné posouvat na požadovanou pozici.

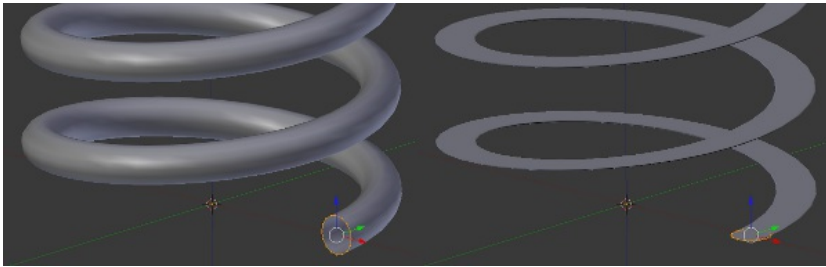
Modifikátor zkroucení

Mód: libovolný režim

Panel: Modifiers

## Popis

Modifikátor zkroucení (Screw) je podobný nástroji Screw ze sady nástrojů (Tool Shelf) v tom, že bere profilový objekt (sít Mesh nebo křivku Curve) a vytváří jeho rotací a posuvem objekt spirálovitého tvaru.



Důležité je správně zarovnat profilový objekt.

Profil by měl být dvojrozměrný a odpovídajícím způsobem zarovnaný ke hlavnímu směru působení modifikátoru.

## Volby



Modifikátor Screw

Axis

Osa podél níž vznikne spirála.

Screw

Výška jedné otočky spirály.

AxisOb

Jméno objektu definujícího směr osy.

Object Screw

Pro definici hodnoty Screw použijte Axis Object.

Angle

Stupňů na jednu otáčku.

Steps

Počet kroků použitých pro jednu otáčku (zobrazeno v pohledu 3D).

Render Steps

Stejně jako výše je využito během renderování. Zvýšená zvyšuje kvalitu výsledku a výpočetní náročnost.

Calc Order

Pořadí hran během výpočtu zamezuje problémům s normálami. Potřebné pouze pro sítové objekty, nikoli křivky.

Flip

Obrácení normálního směru.

Iterations

Počet otáček.

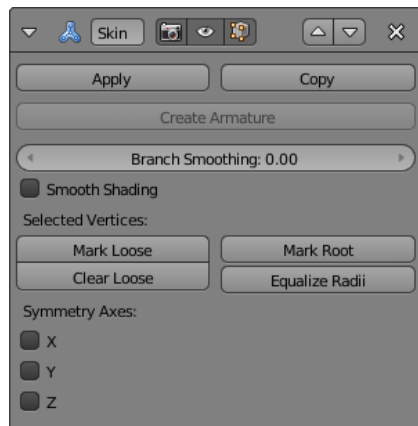
Modifikátor kůže (Skin)

Mód: jakýkoli

Panel: Modifiers

Modifikátor Skin (kůže) používá vrcholy a hrany objektu k vytvoření povrchu podobného kůži. Realističnost povrchu lepší definicí tvaru je možné získat nastavením optimálního vrcholového poloměru. Je to velice rychlý způsob, jak vytvořit základní povrchové síť pro modelování organických tvarů s libovolnou topologií. Stěny jakožto vstup modifikátoru jsou procesem modifikace ignorovány.

## Volby



Rozhraní modifikátoru Skin.

Create Armature

vytvoření armatury požadovaného objektu.

Branch Smoothing

vyhladí protínající se hrany.

Selected Vertices

vybrané hrany

Mark Loose

Označí vybraných hrany jakožto ztrátových bodů. Ztráty definují body, které se vnoří ve své stěny a vzájemně vyplní mezery.

Clear Loose

Vynechat ztráty. Počáteční chování .

Mark Root

Označení kořenenových bodů objektu. Jsou centrem objektu.

Equalize Radii

Makes the skin radii of selected vertices equal on each axis.

Vytvoří poloměry vybraných hran pro každou osu.

## Příklad

- Vyberte krychli. Přepněte pomocí ⇐ Tab do editačního režimu a spojte všechny osy do jednoho bodu pomocí AltM » At Center. Klávesou E pak Z dojde k vytlačení bodů podél osy Z.

Nastavení příznaku - Skin Node Set Flag

Jeden z vrcholů síťoviny musí být nastaven jako kořenový (Root). Pokud jste náhodou odstranili výchozí kořenový vrchol, vyberte jiný stiskem tlačítka Skin Node Set Flag a Mesh Tools, Nabídka nastavení nového vrcholu jako kořene.



Jednoduchá kreatura vytvořená pouze pomocí modifikátoru Skin.

- V menu modifikátorů přidejte modifikátor Skin.
- Přepněte ⇐ Tab do režimu editace a začněte s vytahováním. Pomocí Z je možné vidět aktuální "Z-koule", změna wireframe mode. Tyto koule jsou aktuální prvky s mnoha polygony, a tak se může projevit nižší výkon počítače.

- Snažte se dostat náčrtu výsledky podobné obrázku (jednoduchá kreatura, vyrobená pouze modifikátem Skin), a to prostřednictvím vytlačování vrcholů objektu .
- Použitím CtrlA změňte velikost jednotlivých oblastí vnitřku kreatury.
- Použitím Mark Loose v oblastech, jako je krk sloučíte tyto plochy více dohromady.
- Pro získání hladšího výsledku aktivujte hladké stínování (Smooth Shading) a použijte na objekt Ctrl3.

## Externí odkazy

- [Skin Modifier Development at Blender Nation](#) — První demonstrace modifikátoru od Nicholase Bishopa (Březen 2011)
- Ji, Zhongping; Liu, Ligang; Wang, Yigang (2010). *B-Mesh: Rychlé modelování povrchu*. *Computer Graphics Forum* 29(7), pp. 2169-2178. — *The work this modifier is based on* (<http://www.math.zju.edu.cn/ligangliu/CAGD/Projects/BMesh/Paper/BMesh.pdf> *direct link to PDF*)
- [Related thread on Blender artists](#)

Modifikátor armatury, vyztužení

Mód: Object mode

Panel: Modifiers

## Popis

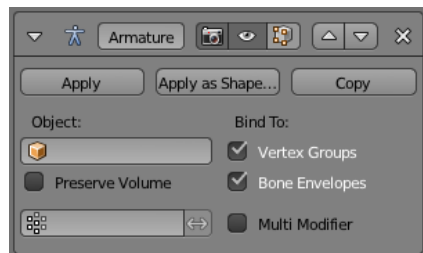
ověřeno, vyzkoušeno. Doplnit o množiny vrcholů a jejich seskupení...

The Armature modifier is used for building skeletal systems for animating the poses of characters and anything else which needs to be posed. With the Vertex Group and Multi Modifier options, you can use several armatures to animate a single (mesh) object.

By adding an armature system to an object, that object can be deformed accurately so that geometry doesn't have to be animated by hand. The Armature modifier allows objects to be deformed by bones simply by specifying the name of the armature object, without having to use the (old) "parent/child" system.

For more details on armatures usage, see [this chapter](#).

## Nastavení



Modifikátor Armature

### Object

The name of the armature object used by this modifier.

### Preserve Volume

Use quaternions to get smoother and nicer rotation interpolations.

### Vertex Group

The name of a vertex group of the object, the weights of which will be used to determine the influence of this Armature modifier's result when mixing it with the results from other Armature ones. Only meaningful when having at least two of these modifiers on the same object, with Multi Modifier activated.

### Multi Modifier

Use the same data as a previous (Armature?) modifier as input. This allows you to use several armatures to deform the same object, all based on the "non-deformed" data (i.e. this avoid having the second Armature modifier deform the result of the first one...). The results of the Armature modifiers are then mixed together, using the weights of the VGroup vertex groups as "mixing guides".

### Bind To

Method to bind the armature to the mesh.

### Vertex Groups

Enable/Disable vertex groups defining the deformation (i.e. bones of a given name only deform vertices belonging to groups of same name).

### Bone Envelopes

Enable/Disable bone envelopes defining the deformation (i.e. bones deform vertices in their neighborhood).

### Invert

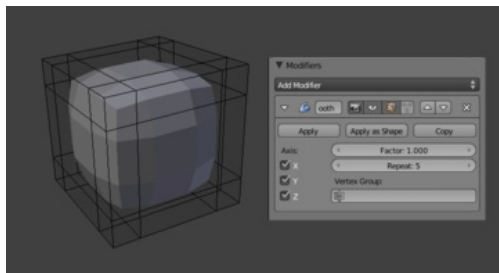
Inverts the influence set by the vertex group defined in previous setting (i.e. reverts the weight values of this group).

Modifikátor vyhlazení (Smooth)

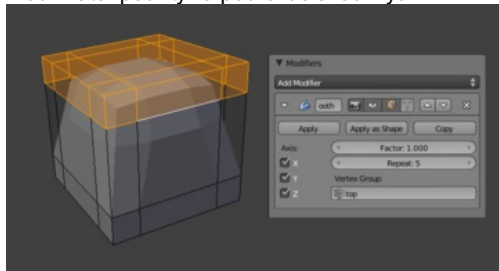
Mód: jakýkoli

Panel: Modifiers

## Popis



Modifikátor použitý na podrozdělenou krychli



Stejně jako výše pouze na výběr skupiny vrcholů

Tento modifikátor vyladí

Tento modifikátor vyhlazuje pletivo zploštěním úhlů mezi sousedními plochami stejně jako v kontextu Smooth při Editaci. Vyhlazuje povrch bez dalšího rozdělování síťoviny a tedy zvyšováním celkového počtu vrcholů objektu.

Tento modifikátor není omezen jen na vyhlazení. Její řídicí faktor může být nastaven mimo rozsah **0 - 1** (včetně záporných hodnot), což může dát vzniku zajímavých deformací sítě.

## Volby

X, Y, Z

Přepínací tlačítka směru pro jednotlivé osy X, Y a Z.

Factor

Faktor ovlivňující sílu vyhlazení. Rozsah je od **0.0** do **1.0** (**0.0**: vypnuto, **0.5**: stejně jako tlačítko Smooth, **1.0**: maximální). Hodnoty mimo tento rozsah způsobí pokrivení síťoviny.

Repeat

Počet iterací vyhlazení. Je ekvivalentní několikanásobnému stisku tlačítka [Smooth](#).

Vertex Group

Název skupiny vrcholů, na které modifikátor působí. To umožňuje selektivní vyhlazování v reálném čase s využitím vah jednotlivých vrcholů.

## Kolize

Se síťovými obkety mohou kolidovat částice ([Particles](#)), [Soft Bodies](#) a objekty oblečení ([Cloth objects](#)). Objekty typu Boid ([Boids](#)) se kolizím vyhýbají.

- Objekty potřebují sdílet alespoň jednu společnou vrstvu.
- Je možné omezit vliv částic na skupinu objektů (v panelu tíhové pole [Field Weights](#)).
- *Deflection* for softbody objects is difficult, they often penetrate the colliding objects.
- Částice vlasů (Hair) ignorují deflexi objektů (ale je možné je animovat jako softbodies které již deflexi respektují).

Je možné změnit nastavení deflexe pro objekt. Tím se přepočítají částice, softbody nebo systém oblečení. Je třeba uvolnit keš ((Free Cache), což neprobíhá automaticky. Je možné keš vymazat pro všechny vybrané objekty pomocí zkratky CtrlB → Free cache selected.

Mode: Object Mode

Panel: Object context → Physics sub-context → Collision

## Nastavení

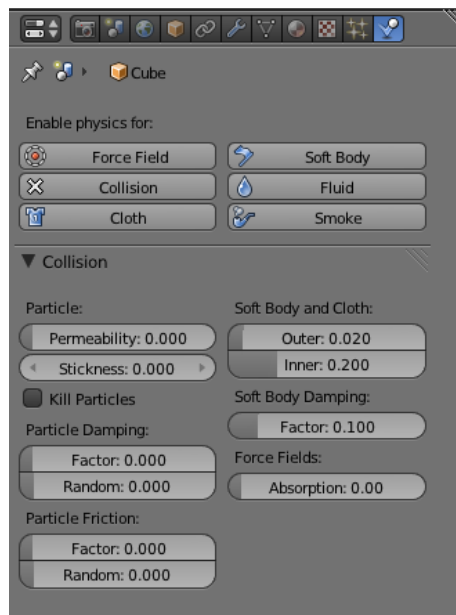


Image 1: Panel Collision v podkontextu Physics.

### Permeability

Fraction of particles passing through the mesh. Can be animated with Object lpos, Perm channel.

### Stickiness

How much particles stick to the object.

### Kill Particles

Deletes Particles upon impact.

### Damping Factor

Damping during a collision (independent of the velocity of the particles).

### Random damping

Random variation of damping.

### Friction Factor

Friction during movements along the surface.

### Random friction

Random variation of friction.

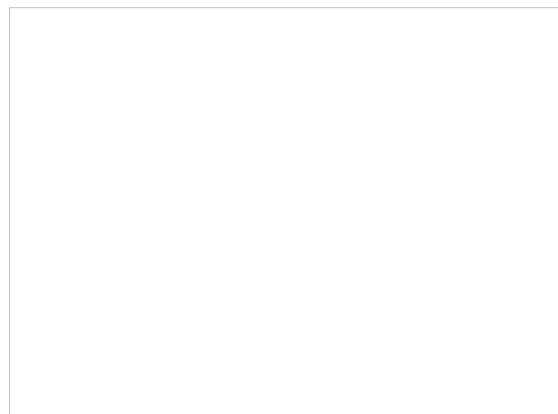


Image 1b: A softbody vertex colliding with a plane.

## Interakce Soft Body a Oblečení

### Outer

Size of the outer collision zone.

### Inner

Size of the inner collision zone (padding distance).

Outside and inside is defined by the face normal, depicted as blue arrow in (*Image 1b*).

### Damping Factor

Damping during a collision.

*Softbody* collisions are difficult to get perfect. If one of the objects move too fast, the soft body will penetrate the mesh. See also the section about [Soft Bodies](#).

## Interakce se silovými poli

### Absorption

A deflector can also deflect effectors. You can specify some collision/deflector objects which deflect a specific portion of the effector force using the Absorption value. 100% absorption results in no force getting through the collision/deflector object at all.

If you have 3 collision object behind each other with e.g. 10%, 43% and 3%, the absorption ends up at around 50% ( $100 \times (1 - 0.1) \times (1 - 0.43) \times (1 - 0.03)$ ).

## Příklady



Image 2: Deflected Particles.

Here is a Meta object, duplivered to a particle system emitting downwards, and deflected by a mesh cube:

## Tipy

- Make sure that the normals of the mesh surface are facing towards the particles/points for correct deflection.
- Hair particles react directly to force fields, so if you use a force field with a short range you don't need necessarily collision.
- Hair particles avoid their emitting mesh if you edit them in Particle mode. So you can at least model the hair with collision.

## Částice (Particles)

Mezi "částice" zahrnujeme velkou spoustu prvků emitovaných ze síťoviny objektů, obvykle v tisících. Každá částice může být světelným bodem nebo sítí a může být dynamicky propojená s ostatními. Na částice může působit mnoho různých vlivů a sil, mohou mít vlastní životnost. Dynamické částice mohou představovat oheň, kouř, mlhu a další materiály jako je prach nebo samotná kouzla.

[Hair](#) type particles are a subset of regular particles. Systém vlasů (hair particles) tvořený prvky, které mohou také představovat vlasy, srst, trávu, či štětiny.

Vidíte částice jako modifikátor Particle, ale všechna nastavení se provádí v záložce Particle tab.

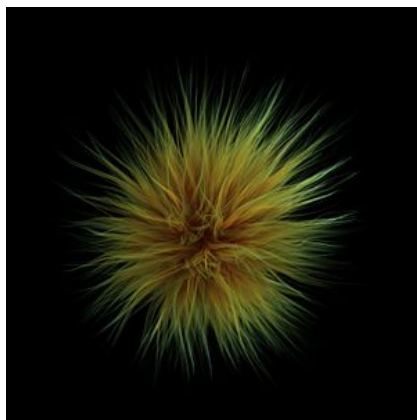


Image 1: Some fur made from particles  
([Blend file](#)).

Částice obecně vytékají ze své rodné sítě do prostoru. Jejich pohyb může být ovlivněn mnohými skutečnostmi, včetně:

- Počáteční rychlost.
- Pohyb emitteru - vysílače (vrchol, stěna, objekt).
- Pohyb v závislosti na "gravitaci" nebo "odporu vzduchu".
- Vliv silových polí, jako je vítr, víry nebo pohyb podél křivky.
- Interakce s jinými předměty, jako jsou kolize.
- Částečně inteligentní chování členů stáda (stádo, školy, ...), které reagují s ostatními členy svého stáda, a zároveň se snaží dosáhnout cíle, nebo vyhnout dravcům.
- Hladký pohyb s pomocí fyziky měkkých částí (pouze částicové systémy typu Hair).
- Nebo dokonce pouze manuální transformace.

## [Lattices](#).

Particles may be rendered as:

- [Halos](#) (for Flames, Smoke, Clouds).
- Meshes which in turn may be animated (e.g. fish, bees, ...). In these cases, each particle "carries" another object.
- [Strands](#) (for [Hair](#), [Fur](#), [Grass](#)); the complete way of a particle will be shown as a strand. These strands can be manipulated in the 3D window (combing, adding, cutting, moving, etc).

Every object may carry many particle systems. Each particle system may contain up to 100.000 particles. Certain particle types (Hair and Keyed) may have up to 10.000 children for each particle (children move and emit more or less like their respective parents). The size of your memory and your patience are your practical boundaries.

## Incompatibility with Prior Versions

There are many differences between the "old" particle system that was used up to and including version 2.45, and the "new" particle system. There are many things possible now that could not be done with the old system. The new system is incompatible to the old system, though Blender tries to convert old particle systems, which works only to some extent. The old system is most like the new Emitter system (keep reading to find out what that is). If you are using an old version of Blender 2.45 and previous, [click here to access the old documentation](#).

## Postup - Workflow

The process for working with standard particles is:

1. Create the mesh which will emit the particles.
2. Create one or more Particle Systems to emit from the mesh. Many times, multiple particle systems interact or merge with each other to achieve the overall desired effect.
3. Tailor each Particle System's settings to achieve the desired effect.
4. Animate the base mesh and other particle meshes involved in the scene.
5. Define and shape the path and flow of the particles.
6. For [Hair](#) particle systems: Sculpt the emitter's flow (cut the hair to length and comb it for example).
7. Make final render and do physics simulation(s), and tweak as needed.

## Vytvoření částicového systému

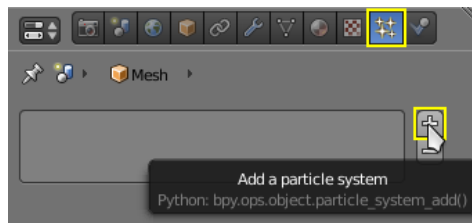


Image 2: Adding a particle system.

To add a new particle system to an object, go to the Particles tab of the object Settings editor and click the small + button. An object can have many Particle Systems.

Each particle system has separate settings attached to it. These settings can be shared among different particle systems, so one doesn't have to copy every setting manually and can use the same effect on multiple objects. Using the Random property they can be randomized to look slightly different, even when using the same settings.

### Typy částicových systémů

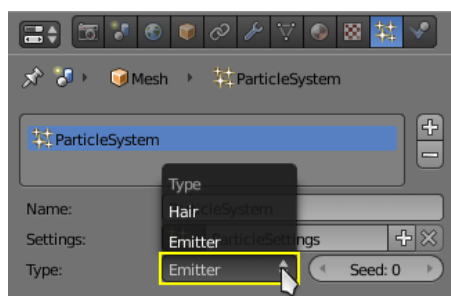


Image 3: Particle system types.

After you have created a particle system, the Property window fills with many panels and buttons. But don't panic! There are two different types of particle systems, and you can change between these two with the Type drop-down list:

#### Emitter

This parallels the old system to the greatest extent. In such a system, particles are emitted from the selected object from the Start frame to the End frame and have a certain lifespan.

#### Hair

This system type is rendered as strands and has some very special properties: it may be edited in the 3D window in realtime and you can also animate the strands with [Cloth Simulation](#).

The settings in the Particle System panel are partially different for each system type. For example, in *Image 3* they are shown for only system type Emitter.

### Společné volby

Each system has the same basic sets of controls, but options within those sets vary based on the system employed. These sets of controls are:

<a href="#">Emission</a>	Settings for the initial distribution of particles on the emitter and the way they are born into the scene.
<a href="#">Cache</a>	In order to increase realtime response and avoid unnecessary recalculation of particles, the particle data can be cached in memory or stored on disk.
<a href="#">Velocity</a>	Initial speed of particles.
<a href="#">Rotation</a>	Rotational behavior of particles.
<a href="#">Physics</a>	How the movement of the particles behaves.
<a href="#">Render</a>	Rendering options.
<a href="#">Display</a>	Realtime display in the 3D View.
<a href="#">Children</a>	Control the creation of additional child particles.
<a href="#">Field Weights</a>	Factors for external forces.
<a href="#">Force Field</a>	Makes particles force fields.
<a href="#">Settings</a>	
<a href="#">Vertex Groups</a>	Influencing various settings with vertex groups.

## Odkazy

- [Tutorials](#)
- [Physics Caching and Baking](#)
- [Particle Rewrite Documentation](#)
- [Thoughts about the particle rewrite code](#)
- [Static Particle Fur Library](#)

## Úvod

Osvětlení je velice důležitou oblastí renderování. Je zhruba na stejné úrovni jako modelování, materiály a textury. I ta nejlépe vymodelovaná a texturami obalená scéna bude ve výsledku špatná bez odpovídajícího osvětlení.

## Omezení a vlivy pohledu

Barva objektu a osvětlení scény je ovlivněno těmito skutečnostmi:

- Vaší schopností vnímat rozdílnosti barev
- Médium skrze které sledujeme obraz (LCD panel, tištěný obraz,...)
- Kvalita obrázku (například JPEG s kompresí **0.4** versus **1.0**).
- Prostředí ve kterém sledujeme obrázek (například klasická monitor v temné místnosti, nebo v osvětlené místnosti)
- Schopnosti mozku vnímat intenzitu a barvu objektu v závislosti na barvě pozadí, které lze měnit pomocí technik Blenderu například v [Kompozitní nody](#).

## Globální vlivy

In Blender, the elements under your control which affect lighting are:

- The color of the world [ambient light](#).
- The use of [Ambient Occlusion](#) as a way to cast that ambient light onto the object.
- The degree to which the ambient light colors the [material](#) of the object.
- The use of [indirect lighting](#), where the color of one object radiates onto another.
- The render engine used (Blender Internal versus [Yafray](#)).
- The [lamps](#) in your scene.

The physics of light bouncing around in the real world is simulated by Ambient Occlusion (a world setting), buffer shadows (which approximate shadows being cast by objects), ray tracing (which traces the path of photons from a light source). Also, within Blender you can use [indirect lighting](#). Ray tracing, ambient occlusion, and indirect lighting are computer-intensive processes. Blender can perform much faster rendering with its internal scan line renderer, which is a very good scan line renderer indeed. This kind of rendering engine is much faster since it does not try to simulate the real behavior of light, assuming many simplifying hypotheses.

## Nastavení osvětlení

Only after the above global influences have been considered, do you start adding light from lamps in your scene. The main things under your control are the:

- Type of light used (Sun, Spot, Lamp, Hemi, etc.).
- Color of the light.
- Position of the light and its direction.
- Settings for the light, including energy and falloff.

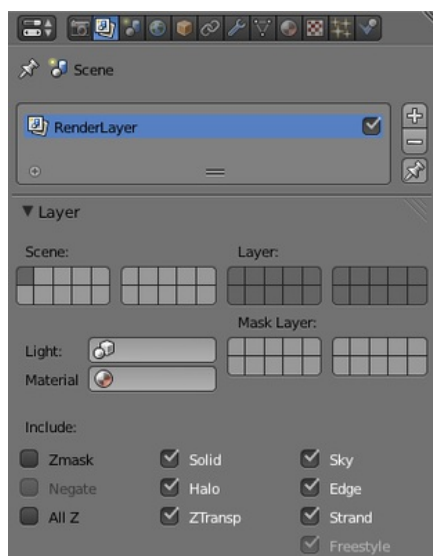
Then you are back to how that material's [shader](#) reacts to the light.

This chapter attempts to address the above, including how lights can work together in rigs to light your scene. In this chapter we will analyze the different type of lights in Blender and their behavior; we will discuss their strong and weak points. We also describe many lighting rigs, including the ever-popular three-point light method.

## Lighting in the Workflow

In this user manual we have placed Lighting before Materials; you should set up your lighting before assigning materials to your meshes. Since the material shaders react to light, without proper lighting, the material shaders will not look right, and you will end up fighting the shader, when it is really the bad lighting that is causing you grief. All of the example images in this section do not use any material setting at all on the ball, cube or background.

## Overriding Materials to Reset Lighting



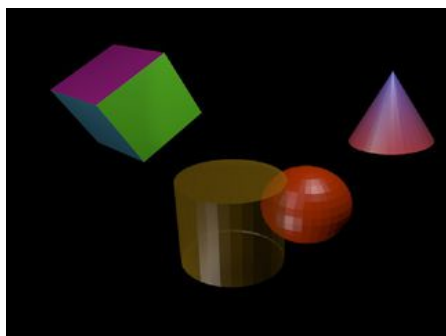
Material field in the Render Layers panel

If you have started down the road of assigning materials, and are now fiddling with the lighting, we suggest that you create a default, generic gray material—no Vertex Color, no Face Texture, no Shadeless, just plain old middle gray with RGB of **(0.8, 0.8, 0.8)**. Name this “Gray”.

Next go to the Render context. In the Render Layers panel, select your new “Gray” material in the Material field. This will override any materials you may have set, and render everything with this color. Using this material, you can now go about adjusting the lighting. Just empty this field to get back to your original materials.

## Úvod do materiálů

Materiál určuje umělecké kvality substance objektu, z něhož je tvořen. V nejjednodušším případě je možné využít materiálu pro demonstraci, z čeho je vytvořen, nebo prostě "vybarvit" objekt, různou barvou. Obvykle je reprezentace materiálu mnohem složitější. Vstupuje do hry celková kvalita povrchu. Tedy jako například barva, svítivost, lesk, odrazivost, průsvitnost, průhlednost. Samozřejmě každá z těchto vlastností je založena na fyzikálních principech vlivu objektu na průchod světla a například průhlednost se dále řídí dalšími principy jakým je index lomu v opticky hustším prostředí.



Různé základní materiály (jednoduchý, vícenásobný, průsvitný, vykreslení hran).

Základní (netexturovaný) materiál je uniformní pro každý povrch objektu (přestože se jednotlivé pixely vykreslení objektu mohou lišit kvůli světelným efektům). Také různé povrchy objektu mohou mít odlišné materiálové vlastnosti. (viz [Násobné materiály](#)).

V Blenderu mohou materiály mít volitelně přidružené textury. Textury popisují substanci, podloží: t.j. povrchovou úpravu, drsnost skla nebo vyřpané linie. Kapitola [Textury](#) popisuje připojení textur k materiálům.

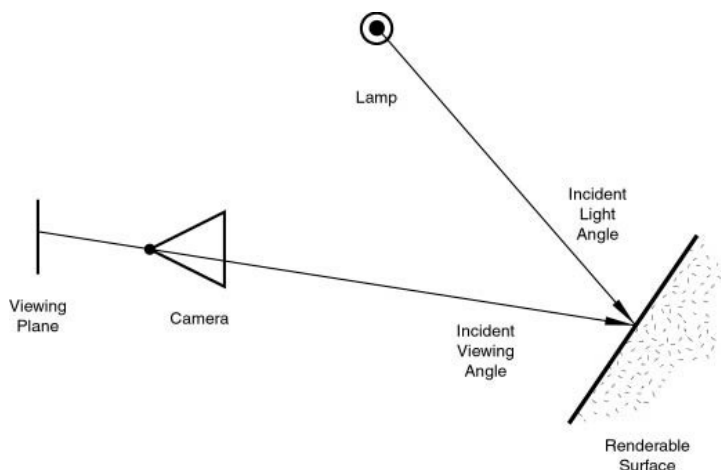
## Jak materiály fungují

Před tím, nežli porozumíte jak efektivně navrhovat materiály, musíte porozumět simulaci světla a vlivu povrchů na něj. Také jak pracuje renderovací engine Blenderu a jak je možné nastavením vlastností materiálů měnit tyto interakce světla a povrchu.

Blenderem vyrenderovaný obraz je projekcí scény na imaginární podložku nazývanou *deska pohledu*. Ta je analogií filmovému okénku tradičního fotoaparátu, nebo tyčinkám a čípkům na sítnici lidského oka pouze s tím rozdílem, že senzory vnímají simulované, nikoli reálné světlo.

Pro renderování obrazu musí být ve scéně jasné, která světla mají vliv na každý vnímaný bod desky pohledu. Nejvýhodnější pro výpočty je otočit směr světelného toku a směřovat jej od světelného čidla (kamery, oka) směrem k objektu ve scéně.

Vlastnosti povrchu a úhlu dopadu světla vypoví mnoho o tom, kolik světla se odrazí do zorného bodu. (*Princip renderovacího engine.*).



Základní princip renderovacího stroje.

Two basic types of phenomena take place at any point on a surface when a light ray strikes it: diffusion and specular reflection. Diffusion and specular reflection are distinguished from each other mainly by the relationship between the incident light angle and the reflected light angle.

The shading (or coloring) of the object during render will then take into account the base color (as modified by the diffusion and specular reflection phenomenon) and the light intensity.

Using the internal ray tracer, other (more advanced) phenomena could occur. In ray-traced reflections, the point of a surface struck by a light ray will return the color of its surrounding environment, according to the rate of reflection of the material (mixing the base color and the surrounding environment's) and the viewing angle.

On the other hand, in ray-traced refractions, the point of a surface struck by a light ray will return the color of its background environment, according to the rate of transparency (mixing the base color and the background environment's along with its optional filtering value) of the material and the optional index of refraction of the material, which will distort the viewing angle.

Of course, shading of the object hit by a light ray will be about mixing all these phenomena at the same time during the rendering. The appearance of the object, when rendered, depends on many inter-related settings:

- Svět (Ambient color, Radiosity, Ambient Occlusion)
- Světla
- Nastavení materiálu (zářivost, svítivost a každé další nastavení)
- Textury a způsob jejich míchání
- Uzly materiálu
- Kamera
- Pozorovací úhel
- Obstructions and transparent occlusions
- Sdíly od dalších odrazivých nebo průsvitných objektů
- Nastavení renderu
- Rozměry objektu (Nastavení SS má vztah k rozměrům)
- Tvar objektu (odrazy,fresnelovy jevy)

## Použití materiálů



### Check your Render

When designing materials (and textures and lighting), frequently check the rendered appearance of your scene, using your chosen render engine/shader settings. The appearance might be quite different from that shown in the texture display in the 3D panel.

As stated above, the material settings usually determine the surface properties of the object. There are several ways in which materials can be set up in Blender. Generally speaking, these are not compatible - you must choose which method you are going to use for each particular object in your scene.

First, you can set the [Properties](#) in the various Material panels.

Second, you can use [Nodes](#); a graphical nodes editor is available.

Last, you can directly set the color of object surfaces using various special effects. Strictly speaking, these are not materials at all, but they are included here because they affect the appearance of your objects. These include [Vertex Painting](#), [Wire Rendering](#), [Volume Rendering](#), and [Halo Rendering](#).

The exact effect of Material settings can be affected by a number of system settings. First and foremost is the Render Engine used - Cycles and the Blender Render Engine (aka Blender Internal or BI) require quite different illumination levels to achieve similar results, and even then the appearance of objects can be quite different. Also, the material properties settings can be affected by the texture method used (Single Texture, Multitexture or GLSL). So it is recommended to always select the appropriate system settings before starting the design of materials.

## Úvod k oboru textury

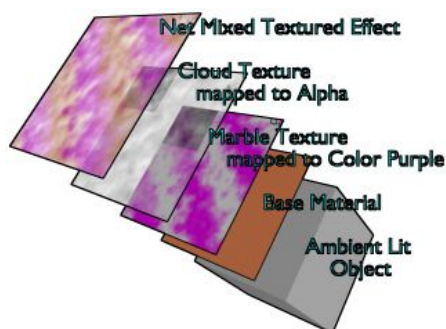
Mapování textur je metoda přidávající další detaily těles projekcí obrazů a vzorů na jejich povrchy. Projecce obrazů a vzorů mohou mít vliv nejenom na barvu, ale také na svítivost, odrazivost, průsvitnost a dokonce třírozměrnou hloubku. Velice často jsou obrazy a vzory promítány na povrchy těles během doby renderování, ale také je toto mapování využíváno pro kreslení, barvení a deformaci objektů.

IV Blenderu mohou být textury:

- aplikovány na *materiál*,
- aplikovány na [pozadí světa](#)
- použity pro *namalování*, jako například:
  - [výtvarný režim](#),
  - [kreslení textury](#).
- přidruženy k modifikátorům, jako:
  - částicové textury,
  - textury oceánů.

## Materiálové textury

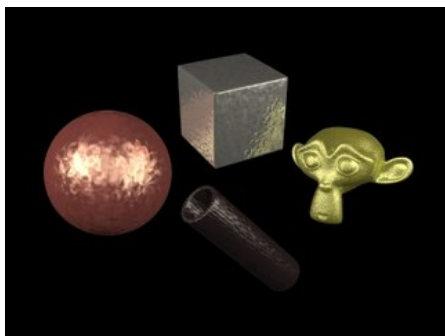
Nastavení materiálu, který vnímáme hladký a "uniformní" neodpovídá zcela realitě, ve které absolutní uspořádanost bývá spíše výjimkou. Blender dává uživatelům možnosti aplikovat tzv. "textury", které mohou měnit odrazivost, lesk a dokonce fyzický povrch materiálu.



Vrstva textury na základním materiálu

Textury se chovají jako další vrstvy položené na základní materiál. Textury mají vliv na více aspektů obarvování objektů. Celková barva objektu závisí na druhu vrstvení efektů jak je ukázáno na obrázku dále. Vrstvy (nezaměňovat s vrstvy modelu) budou:

1. Mějme objekt osvětlený okolím pomocí nastavení "světa".
2. Základní materiál, jehož celý povrch je jednotná barva reaguje na světelné podmínky a díky odrazům je vidět v různých odstínech dané barvy, různých odleskech, rozptylu a zrcadlení.
3. **Základní primární textura** překrývá objekt světle fialovou barvou.
4. **Druhá textura mraků** která způsobuje průhlednost povrchu má vliv na hodnotu Alfa kanálu (průsvitnost).
5. Tyto dvě textury jsou **smíchány** se základním materiálem a vytvoří výsledný jev: krychle nalíčená fialově-hnědou mlhou.



Některé kovové textury

Tato možnost využít *více nežli jednu* texturu k dosažení kombinovaného jevu je jedním z "tajemství", jak vytvořit skutečně realistické objekty. Pokud se pozorně "zadíváte do světa" když zkoumáte jakýkoli skutečný objekt, budete vidět, že výsledná barva je kombinací jiných. Tyto charakteristiky mohou být víceméně silně závislé na různých úhlech, na podmínkách osvětlení a tak dále. Blender umožňuje docílit výsledku více cestami. Můžeme použít "zásobník vrstev textur" jak je popsáno v [této kapitole](#), nebo je možné využít velice komplexních sítí ("nudlí"... ) z "texturových uzlů" jak je diskutováno [zde](#); volba je jen na Vás.

Materiály a textury se rozpadají do těchto hlavních kategorií:

### Procedurální textury

Jsou textury generované pomocí matematických vzorců. Například dřevo-Wood, mraky-Clouds a vířivý šum-Distorted Noise

### Obrázky a filmy

Fotografie a filmy je možné projektovat na povrchy objektů. Například plochá mapa Země namapovaná na kouli.

### Mapy prostředí - Environment Maps

Textury použité pro vytvoření dojmu odrazů a lomu. Například obrázků ulice zrcadlený oknem automobilu.

### Datové nebo Modifikované Textury

Textury získané z kořenových dat nebo získané určitým modifikátorem ve scéně.

Například:

- volumetrické (objemové) materiály využívají textury typu Voxel Data, nebo textury bodových hustot (Point Density)
- textury získané z modifikátoru oceán.

poznámka k editorům, a překladatelům

je vhodné dále zpracovat více popisu pro všechny typy textur.

## World Textures

Todo - dodělat ang. verzi

## Brush Textures

Todo - dodělat ang. verzi

## Svět - World



### World panel

Blender dává uživateli mnoho zajímavých nastavení pro kompletaci scény přidáním pozadí a některých zajímavých "hloubkových" efektů. Všechny jsou přístupné pomocí kontextu World. Přednastavený "svět" je velice strohý, uniformní. Lze jej změnit, nebo přidat nový svět.

Máte:

#### [Pozadí - Background](#)

Barva a textura pozadí světa se speciálním nastavením mapování souřadnic.

#### [Mist](#)

Přidat mist do vaší scény pro zlepšení pocitu hloubky pohledu.

#### [Hvězdy - Stars](#)

Náhodně uzavřít svět do teček hvězd s různým vyzařováním.

Zatímto tato nastavení jsou jednoduchým způsobem přidání scénické efekty, je možné použít daleko složitějších, nejšťastěji preferovaných [kompozitních nódů](#). Například filtrování hodnoty Z (vzdálenost od kamery) nebo normál (směry povrchů) skrze kompozitní nody mohou mnohonásobně zvýšit pocit hloubky a rozměru a věrnosti zobrazovaného prostoru.

#### Poznámka

Některá nastavení v panelu World spadají do oblasti světla a jsou popsány například v kapitolách [osvětlení](#), kapitola [okolní světlo - Ambient Light](#), [Expozice](#) a [Ambient Occlusion](#). Pro použití slunečního osvětlení (Sun Lamp) a nebe a atmosféru je k dispozici menu Lamp.

Hvězdy

## Popis

Hvězdy jsou náhodně umístěvané objekty vyzařující style "hallo" umístěné v pozadí.

## Volby



Star panel

Size - velikost

Aktuální velikost efektu halo. Je vhodné udržet proměnnou malou, kdy efekt jwe více realističtější.

Colors - barvy

Přidá náhodnou barevnost hvězd narozdíl od čistě bílé barvy.

Min. Dist - min. vzdálenost

*Minimální* vzdálenost hvězd od kamery. Tato vzdálenost by měla být větší, nežli celková velikost scény (pokud samozřejmě nechceme, aby se nám hvězdy nereálně míchaly do popředí scény).

Separation - separace

Toto je *průměrná* vzdálenost hvězd mezi sebou, tedy průmater oddělenosti. Hvězdy totiž mají stále vlastnosti umístění 3D objektu v prostoru, zatímco pozadí oblohy je jejich 2D projekce.

## Úvod

V okamžiku vytváření scénérie s postavami a animacemi je nutné vytvořené postavy, nebo objekty "rozpohybovat". **Rigging** je proces připojení kostry (skeletonu) k tělu jakožto nadřazeného objektu a pomocí této kostry je možné objekt (tělo) deformovat různými způsoby a je možné s objektem pohybovat pomocí pozicování (posing) v prostoru.

Tyto aktivity v zásadě nemění strukturu sítě (mesh) objektu.

Následující kroky jsou typické pro aktivity typu **Rigging**:

1. Přidat armaturu, která začíná jedinou kostí.
2. Přidání dalších kostí podle potřeby a s požadovanou provázaností klouby  
Je možné další kosti vytvářet extruzí stávajících.
3. Editace kostí do odpovídajících proporcí objektu
4. Využití omezovačů  
(t.j. lidský loket povoluje natočení připejovaných kostí v jedné rovině pouze v rozmezí 170 stupňů)
5. Nastavit ostatní 'rest' pozice skeletonu.
6. Aplikovat síťovinu na tuto armaturu (skinning, povrchování, potažení)
7. Definice způsobu vlivu pohybu armatury na povrch  
(folding, flexing, bulging)
8. Nastavit pozice armatury.  
(Zde je více metod: aranžování každé kosti ručně, nebo kopírováním vzoru armatury, nebo aranžováním kostí následující křivky, nebo celkovým následováním pohybu armatury vexterních pohybových dat.
9. Kontrola vlivu pohybu armatury na povrch (kůži), adkustace parametrů.  
(Případně doladění parametrů topologie tak, aby výsledek odpovídal požadovanému)

Všechny tyto kroky jsou vysvětleny detailně dále.

## Armatury

Armatury podobně jako ve skutečnosti mají úkol nejen držet strukturu tělesa pohromadě (například socha), ale také mají podobně jako kosti úkol tělesa uvádět do pohybu.

### Panely armatury a kosti

ukazují jak využít různé panely v Blenderu pro nastavení vlastních armatur a kostí.

### Kosti

vysvětlují vlastnosti kostí, které jsou základními stavebními prvky armatur.

### Vizualizace

dávají čtyři způsoby zobrazení kostí.

### Struktura

vysvětluje strukturu kostí v armatuře.

### Vybírání

vybere pouze část armatury, která odpovídá Vaším požadavkům.

## Editování

### Kosti

Naučit se prakticky upravovat kosti v Blenderu a sledovat, co způsobují.

### Kreslení

Použit nástroje pro načrtnutí skeletonu jednoduše načrtnutím jednotlivých kostí.

### Templátování

nabízí ideální cestu pro znovupoužití již vytvořených koster vašich vlastních modelů..

## Skinning-povrchování

Tato kapitola popisuje jak "nablýskat" postavy z daných armatur. Slovo "Skin" v angličtině vyjadřuje kůži, nebo povrch. V těchto procedurách jde také o jakési "potažení" kostry hmotou, která je pokryta kůží.

### Linking Objects to Bones-přivázání objektů na kosti

Jak učiniti z kosti rodiče objektu tak, aby jej plně kontrolovala a řídila. Tento typ vazby se častop používá v simulaci mechanické vazby, například [Newtonovo kyvadlo](#)), nebo tam, kde části sítě nejsou během pohybu armatury deformovány. Vhodné pro modelování hmyzu, kraba a podobně.

### Skinning to Objects' Shapes-povrchování tvarů

Jaký vliv má armatura, jejíž každá kost ovlivňuje specifickou část tělesa a jeho geometrie. Tento typ vazby je využíván v případě, kdy se hmota během pohybu armatury také tvaruje. Typickým příkladem budiž svaly.

### Retargeting-přecílení

Jak použít zachycené pohybové body (motion-capture data) získané z reálného světa (videa) pro napasování požadovaného pohybu. Tato mravenčí metoda také sáhne až na detailní zpracování pohybu po jednotlivých snímcích..

## Posing - pozicování

Pozicování znamená aranžování objektů ve scéně tak, aby bylo dosaženo požadovaného efektu v celkovém výsledku videoklipu. Jde také částečně o vytváření zajímavých kompozic odpovídajících charakteru daných objektů, nebo bytostí. Shlédněte například "řeč těla - body language" které je u sochy "Myslitel" od Augusta Rodina zcela fascinující. Koukněte na wiki odkazy [Myslitel \(en\)](#), nebo [sochař Auguste Rodin \(cz\)](#).



**poznámka překladatele**

Ze svého soukromého pohledu na problematiku vytváření pozic, animací musím připsat tento komentář: Při požadavku na vytvoření pokud možno co nejreálnější vizualizace pohybu osoby, nebo zvířete, je opravdu vhodné provést skutečná pozorování (video, film), trochu se věnovat i studiu dynamiky kresby daných objektů. Je velice obtížné a patří svým způsobem do šuplíku kumštu vytvořit pohyb postavy tak, aby byl reálný. Můžete si všimnout v řadě komerčních filmů určitých nedokonalostí při chůzi, skocích a podobně. Bytosti jakoby byly gumové, pluli v magnetickém poli země, ale nikoli dopadali nohama na tvrdou zem. Na těchto detailech je patrný přístup a zejména pečlivost animátora. Blender je v této oblasti výtečným pomocníkem, ale "duše animátora a jeho citění a postupy" nejsou prakticky nikdy "zdigitalizovatelné".

Pozicování je užíváno zejména u vytváření animací. Například při tvorbě podávajícího tenisty je vhodné jej zachytit v několika důležitých momentech podání: (a) když drží míč a raketu (b) když vyhazuje míč vzhůru, (c) když udeří raketou do míče a (d) když raketa dosáhne nejnižšího bodu pohybu po odpalu. Blender se sám postará o vytvoření interpolovaných snímků a je jen na uživateli provést v případě nutnosti jejich jemné úpravy.

#### [Visualization-vizualizace](#)

popisuje vizuální nástroje použitelné speciálně pro pozicování.

#### [Editing Poses-editování pozic \(póz\)](#)

jak vytvořit pózu, jak ji upravovat, jak zachytit aktuální snímek v různých časových momentech.

#### [Pose Library-knihovna pozic](#)

ukládání často používaných pozic, nebo existujících pozic z jiných armatur a tím zrychlit činnost modeláře - animátora.

#### [Using Constraints-použití omezovačů](#)

jak použít omezovače tak, aby při animaci nevznikly pozice pro objekt neobvyklé.

#### [Inverse Kinematics-IK-inverzní kinematika](#)

vlastnost, při které změnou jedné kosti v armatuře můžete způsobit pohyb ostatních tak, jak je dáno běžnými pohybovými zákony v přírodě. Pokud pohneme s prstem postavy, způsobíme do jisté míry i pohyb celé ruky.

#### [Spline IK](#)

vlastnost ve které je možné přizpůsobit pohyb soustavy kostí podél křivek.

## Armatury

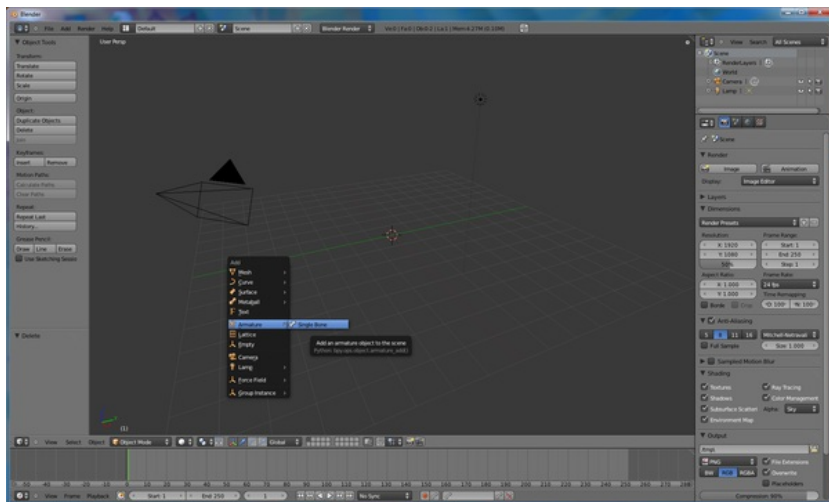
Pojmem "armatura" myslíme typ objektu použitého pro [pozicování](#). Objekty armatury vychází z mnoha námětů samotné skutečné přírody, konkrétně skeletonů, tedy koster (živočichů).

## Vaše první armatura

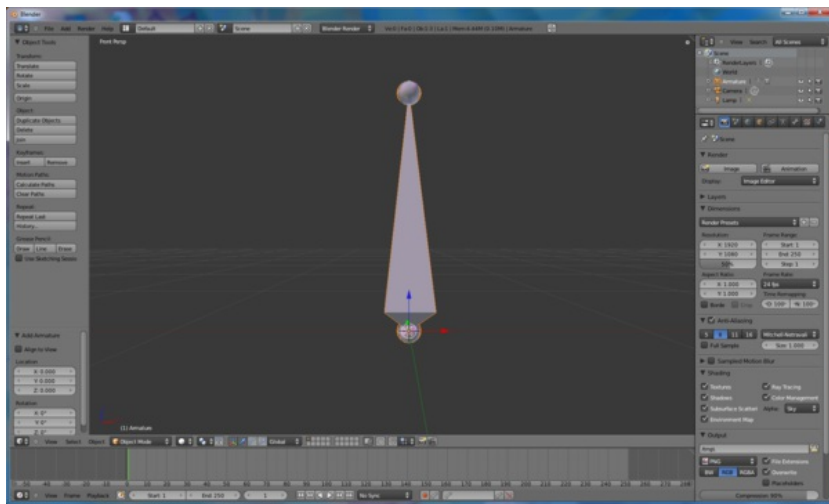
Aby bylo zřejmé, o čem pojednává kapitola armatur, zkusme vytvořit přednastavenou armaturu Blenderu.. (Detailní popisy jsou vysvětleny v [sekci úpravy armatur](#)).

Otevřte obyčejnou scénu, pak:

- vymažte všechny objekty scény
- ujistěte se, že 3D kurzor je v počátku soustavy ⇨ ShiftC
- stiskněte 1 NumPad pro pohled nárysu (čelní pohled)
- pak například:
  - v hlavním menu zvolte přidat: Add > Armature > Single Bone (jednoduchá kost)
  - anebo ve 3D pohledu přidejte armaturu pomocí ⇨ ShiftA » Armature » Single Bone
- stiskněte Del numpad po maximální zvětšení pohledu



Nástroj (Toolbox): Add » Armature » Single Bone



Přednastavená armatura

## Objekt armatura

Jak je vidět, armatura je stejná jako každý jiný objekt v Blenderu:

- má střed, parametry pozice, rotace, zvětšení,
- má vlastní ObData datablok, který lze upravovat v režimu Edit mode.
- Může být přiřazena k jiným scénám a stejně tak mohou dat stejné armatury vystupovat ve více objektech.
- Všechny animace se provádí v režimu Object mode, tedy za pomoci manipulace s celými objekty. Naopak s armaturami manipulujeme v pozičním režimu Pose mode.

Armatury jsou navrženy pro potřeby pozicování, a to ať statické, tak dynamické scény. Mají specifický stav nazývaný "rest position" (původní pozice), což je umístění do "tvary" tak, jak bylo vykresleno v editačním Edit mode.

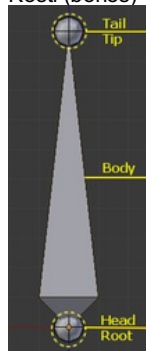
V Edit mode je armatura vždy viditelná v této pozici "rest position", zatímco v režimu Object a Pose mode jde o aktuální "pozici" armatury (dokud není zapnuto tlačítko Rest Position v panelu Armature).

## Přehled kapitoly o armaturách

V detailním popisu armatur se setkáme s těmito sekcemi:

- objekt armatury - [panely](#)
- základy [kostí](#)
- různé způsoby [vizualizace armatur](#)
- [typy struktur](#) armatury
- jak [vybrat](#) její části,
- jak [upravovat armaturu](#),
- jak [upravovat kosti](#),
- jak [upravovat vlastnosti kosti](#),
- jak kreslit armatury pomocí [nástroje "Etch-a-Ton"](#),
- jak využít [šablony \(templates\)](#).

## Kosti (bones)



The elements of a bone.

Kosti jsou základním prvkem objektů.

Kosti mají tři prvky:

- "startovní bod" nazývaný **root (kořen)** nebo **head (hlava)**,
- vlastní tělo "body",
- a "koncový point" nazývaný **tip** nebo **tail**.

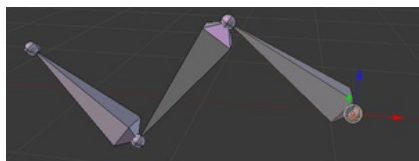
Select the [default armature](#) and press ⇧ Tab to enter Edit mode. As you can see, in this mode you can select the root and the tip, and move them as you do with mesh vertices (don't lose too much time here though, specific pages about selecting and editing will come later).

Both root and tip (the "ends") define the bone by their respective position.

They also have a radius property, only useful for the envelope deformation method (see below).

## Bones Visualization

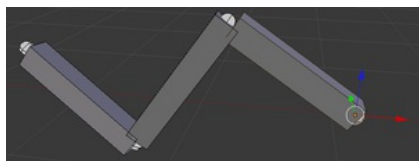
Bones can be visualized in various ways: Octahedron, Stick, B-Bone, Envelope and Wire. Custom shapes can be used, too!



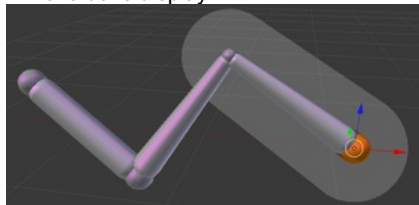
Octahedral bone display.



Stick bone display.



B-Bone bone display.



Envelope bone display.

Since armatures are made of bones, you'll find more about this when we'll talk about [Armatures Visualization](#).

Activating Axes checkmark on the Armature/Display panel, will show local axes for each bone's tip. The Y axis is always aligned along the bone, oriented from root to tip. So, this is the "roll" axis of the bones.



The Bone context.

## Bones properties

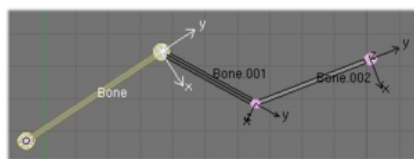
When bones are selected (hence in Edit mode and Pose mode), their properties are shown in the Bone button context of the Properties window.

This shows different panels used to control features of each selected bone, the panels change depending on which mode you're working in.

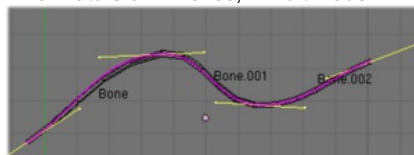
## Bones Rigidity

Even though the bones are rigid (i.e. behave as rigid sticks), they are made out of segments. Segments are small, rigid linked elements that can rotate between each other. By default, each new bone has only one segment and as such it cannot "bend" along its length. It is a rigid bone.

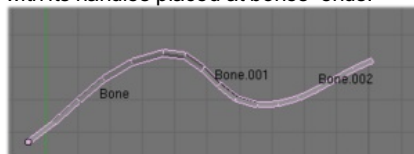
You can see these segments in Object mode and in Pose mode, and only if bones are visualized as B-bones, while in Edit mode bones are drawn as rigid sticks anyway. Note that in the special case of a single bones, you can't see these segments in Object mode, because they're aligned.



An armature of B-Bones, in Edit mode



The Bézier curve superposed to the chain, with its handles placed at bones' ends.



The same armature in Object mode

When you connect bones to form a [chain](#), Blender calculates a Bezier curve passing through all the bones' ends, and bones' segments in the chain will bend and roll to follow this invisible curve.

*You have no direct access to this curve*, you can only control it to some extent using bone properties, as explained in the [editing pages](#).

In An armature of B-Bones in Edit mode we connected 3 bones, each one made of 5 segments. These are B-bones but as you see, in Edit mode they are shown as rigid elements. Look at The same armature in Object mode: now, in Object mode, we can see how the bones' segments smoothly "blend" into each other, even for roll.

Of course, a geometry influenced by the chain is smoothly deformed accordingly to the Bezier curve! In fact, smooth bones are an easy way to replace long chains of many small rigid bones posed using IK...

However, if the chain has an influence on objects rather than geometry, the segments orientation is not taken in account (details are explained in the [skinning part](#)).

When not visualized as B-Bones, bones are always shown as rigid sticks, *even though the bones segments are still present and effective* (see [skinning to ObData](#)).

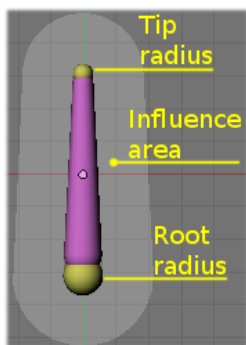
This means that even in e.g. Octahedron visualization, if some bones in a chain have several segments, they will nonetheless smoothly deform their geometry...

## Bones influence

Basically, a bone controls a geometry when vertices "follow" the bone. This is like how the muscles and skin of your finger follow your finger-bone when you move a finger.

To do this, you have to define **how much** a bone influences a certain vertex.

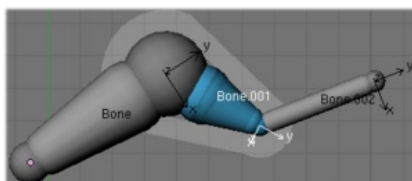
The simplest way is to have each bone affecting those parts of the geometry that are within a given range from it. This is called the *envelope technique*, because each bone can control only the geometry "enveloped" by its own influence area.



A bone in Envelope visualization, in Edit mode.

If a bone is visualized as Envelope, in Edit mode and in Pose mode you can see the area of influence, which depends on:

- the distance property
- the root's radius and the tip's radius.

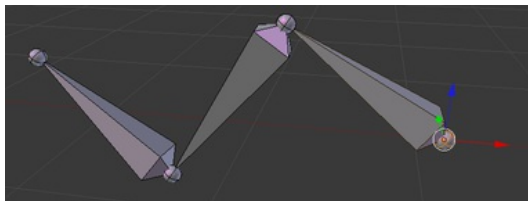


Our armature in Envelope visualization, in Pose mode.

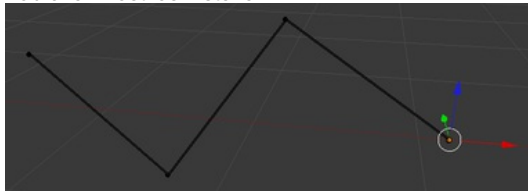
All these influence parameters are further detailed in the [skinning pages](#).

## Vizualizace aramatury

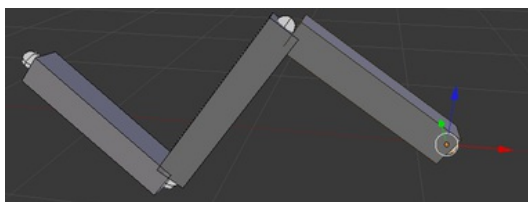
V Blenderu je 5 základních typů vizualizací kostí: Osmistěn (Octahedral), tyčka (Stick), B-Bone, Obálka (Envelope) a drát (Wire). Chování kostí je samozřejmě nezávislé na typu zobrazení. Typ zobrazení pouze umožňuje vhodnější zobrazení jako celku. Záleží na zvyklostech a preferencích modeláře.



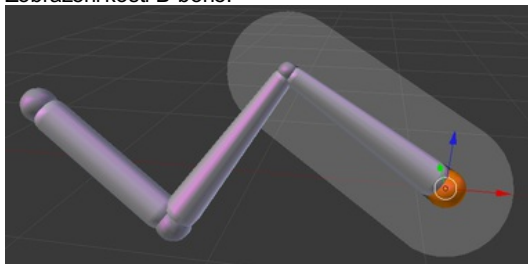
Zobrazení kosti osmistěnem.



Zobrazení kosti tyčkou.



Zobrazení kosti B-bone.



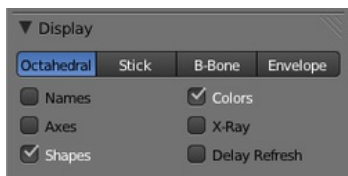
Zobrazení kosti obálkou.

## Panel zobrazení

Mód: režimy Object, Edit a Pose

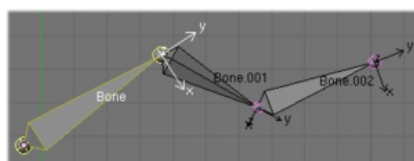
Panel: Display Object Data context

Nejprve zhrňme některé základní obecné vlastnosti vizualizací armatur, viditelné na panelu Display v kontextu dat objektu.



Panel display panel.

## Typy kostí



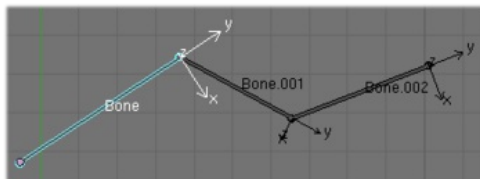
Základní armatura jako osmistěn (Octahedron) v režimu editace (Edit).

Pozn. jde o 40° rolovanou kost Bone.001.

### Kost osmistěn

Toto je přednastavená vizualizace kostí. Většinou vyhovuje pro všechny práce s kostmi a armaturami. Zohledňuje:

- Kořen kosti (bone root - "big" end) a výběžek (tip - "small" end).
- "Velikost" kosti (tloušťka je úměrná délce).
- Rolování (natočení) kosti (díky čtvercovému profilu je patrné natočení kosti).

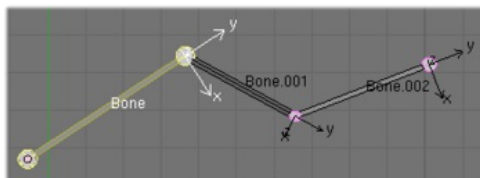


Stejná aramtura ve vizualizaci Stick, v požíčním Pose mode.

Poznámka: Úhel natočení roll u kosti Bone.001 není vidět (vyjma jeho XZ os).

### Kost tyčka

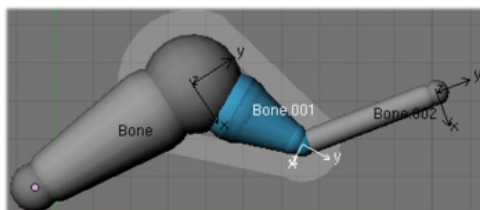
This is the simplest and most non-intrusive visualization. It just materializes bones by sticks of constant (and small) thickness, so it gives you no information about root and tip, nor bone size or roll angle.



The same armature in B-Bone visualization, Edit mode.

### Kost B-Bone

Tato vizualizace představí vícesegmentové kosti jakožto hladké kčivky. Detaily je možné shlédnout na [stránce kostí](#).



Panel skupiny Bone Groups.

### Kost obálka

Toto zviditelnění kostí představí oblasti vlivu deformace pro jednotlivé kosti. Více detailů o zobrazení a vlivu najdete na [stránce vlivu kostí](#).

## Atributy

#### Names

Pokud je povoleno, je vykreslováno jméno každé kosti.

#### Barvy (Colors)

Nastavení je relevantní pro poziční režim modelování (Pose mode) a je detailně úpopsáno [zde](#).

#### Osy (Axes)

Pokud je zaškrtnuto, budou pro každou kost zobrazeny osy lokálního souřadného systému (platí pro režimy editace (Edit mode) a pozicování (Pose mode)).

#### Rentgen (X-Ray)

Pokud je povoleno, kosti získají vlastnost být viditelné i skrze materiál, který je obaluje (sítě - meshes, povrchy - surfaces, ...). To znamená, že jsou vždy viditelné, a tím i byratelné. (Toto je stejná volba jako v panelu zobrazení Display v kontextu dat objektu (Object data). Jde o velice užitečnou funkci v případě modelování mimo drátový (Wireframe) režim.

#### Tvary (Shapes)

Pokud je povoleno, standardní tvar kosti je nahrazen tvarem vybraného objektu v režimech Object a Pose (detaily jsou popsány [níže](#)).

#### Zpoždění obnovy (Delay Refresh)

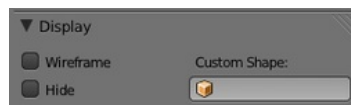
Pokud je povoleno, kosti nebude mít vliv na své potomky během manipulace v pozičním režimu (pose mode).

## Tvary kostí

Mód: Object a Pose režimy

Panel: Panel Display v kontextu Bone.

Blender umožňuje nastavit každé kosti v armatuře specifický tvar (v režimech Object a Pose) za pomoci jiného objektu jako "template". Nejprve je nutné povolit vlastnost pomocí tlačítka Tvary (Shapes) v panelu (Armatury).



Panel Display.

### = Atributy

Wireframe

Pokud je povoleno, kost je zobrazena v dátovém zobrazení v souladu s režimem vykreslování pohledu..

Zkrýt (Hide)

Kost není viditelná pokud není v editačním režimu (Edit mode).

Vlastní tvar (Custom Shape)

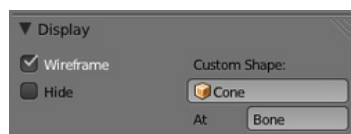
Objekt, který definuje uživatelský tvar vybrané kosti.

Custom At

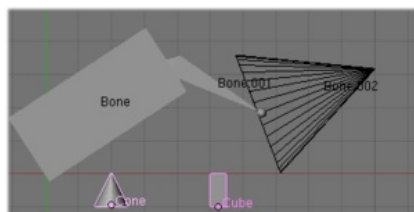
Kost, která určuje zobrazení transformací tvaru kosti

Pro přiřazení žvatelského tvaru kosti je nutné:

- Přepnout do požičního (Pose) režimu pomocí (Ctrl+ Tab).
- Vybrat odpovídající kosti (Kliknutím RMB na kost).
- Jít na panel Display, pole Custom Shape a vybrat objekt 3D, který byl dříve ve scéně vytvořen. V tomto případě používáme krychli a hranol. Je možné volitelně nastavit pole At pro další kost.

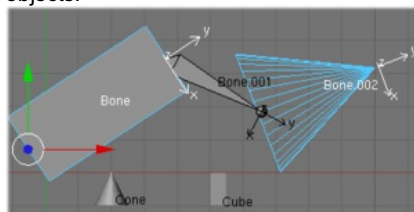


Panel Display.



The armature with shapes assigned to two bones, in Object mode.

Note the centers of the Cone and Cube objects.



The same armature in Pose mode...

Note that:

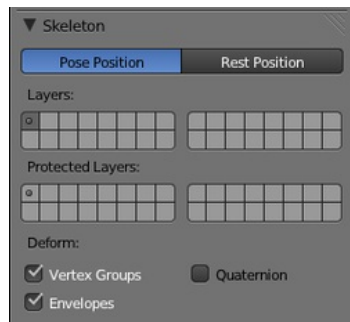
- These shapes will never be rendered – like any bone, they are only visible in 3D views.
- Even if any type of object seems to be accepted by the OB field (meshes, curves, even metas...), only meshes really work – all other types just make the bone invisible; nothing is drawn...
- The center of the shape object will be at the *root of the bone* (see the [bone page](#) for root/tip).
- The object properties of the shape are ignored (i.e. if you make a parallelepiped out of a cube by modifying its dimensions in Object mode, you'll still have a cube shaped bone...).
- The "along bone" axis is the Y one, and the shape object is always scaled so that one Blender Unit stretches along the whole bone length.
- If you need to remove the custom shape of the bone, just right click in the Custom Shape field and select Reset to default value in the popup menu.

So to summarize all this, you should use meshes as shape objects, with their center at their lower-Y end, and an overall Y length of **1.0** BU.

## Armature Layers

Mode: Object, Edit and Pose modes

Panel: Skeleton panel, Object data context



The Skeleton panel.

Each armature has 32 “Armature layers” which allow you to organize your armature by “regrouping” sets of bones into layers; this works similar to scene layers (those containing your objects). You can then “move” a bone to a given layer, hide or show one or several layers, etc.

### Showing/hiding bone layers

Only bones in active layers will be visible/editable – but they will always be effective (i.e move objects or deform geometry), whether in an active layer or not. To (de)activate a layer, you have several options, depending in which mode you are in:

- In all modes, use the row of small buttons at the top of the Display Options group, Armature panel. If you want to enable/disable several layers at once, as usual, hold **⇧ Shift** while clicking...
- In Edit and Pose modes, you can also do this from the 3D Views, either by using the menu (Armature » Switch Armature Layers or Pose » Switch Armature Layers), or the **⇧ ShiftM** shortcut, to display a small pop-up dialog containing the same buttons as described above (here again, you can use **⇧ Shift LMB** clicks to (de)select several layers at once).

### Protected Layers

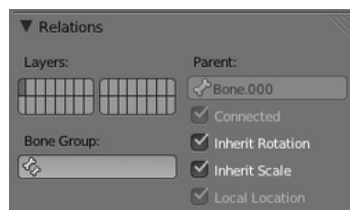
You can lock a given bone layer for all [proxies](#) of your armature, i.e. all bones in this layer won't be editable. To do so, in the Skeleton panel, **Ctrl LMB** click on the relevant button, the layer lock will be enabled.

Protected layers in proxy are restored to proxy settings on file reload and undo.

## Vrstvy kostí

Mód: Object, Edit a Pose módy

Panel: Relations panel Bone kontext



The Relations panel.

### Přesun kostí mezi vrstvami

Obvykle...

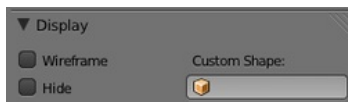
Obviously, you have to be in Edit or Pose modes to move bones between layers – note that as with objects, bones can lay in several layers at once, just use the usual **⇧ Shift LMB** clicks... First of all, you have to select the chosen bone(s)!

- In the Button window, use the “layer buttons” of each selected bone “sub-panel” (Armature Bones panel) to control in which layer(s) it lays.
- In the 3D View window, use the menu (Armature » Move Bone To Layer or Pose » Move Bone To Layer) or hit **M** to show the usual pop-up layers dialog. Note that this way, *you assign the same layers to all selected bones*.

## Skrývání kostí

Mód: Edit a Pose módy

Panel: Display panel, Bone kontext



Panel Display.

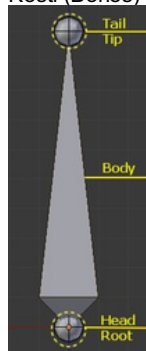
Pro skrývání některýcj kostí není nutné využívat další vrstvu. Stejně jako s objekty, vrcholy nebo řidicími body je možné použít zkratku H pro

- H schová všechny vybrané kosti.
- $\text{Shift}H$  schová všechny kosti vyjma vybraných.
- $\text{Alt}H$  odkryje všechny schované kosti.

Je také možné využít zaškrtnutí boxu Hide na panelu Display v kontextu kosti Bone.

Skryté kosti náleží danému módu. Pokud skryjeme některé kosti v režimu editace Edit mode, budou stále viditelné v pozičním Pose mode režimu a naopak. Ukrytá kost v pozičním režimu Pose mode je také neviditelná v objektovém Object módu. A v Edit mode musí být kost pro ukrytí plně vybrána, nestačí jen její kořenová část...

## Kosti (Bones)



The elements of a bone.

Bones are the base elements of armatures.

They have three elements:

- the “start point” named **root** or **head**,
- the “body” itself,
- and the “end point” named **tip** or **tail**.

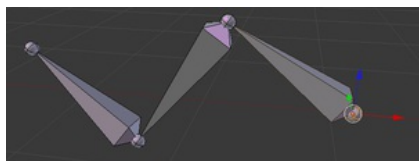
Select the [default armature](#) and press ⇧ Tab to enter Edit mode. As you can see, in this mode you can select the root and the tip, and move them as you do with mesh vertices (don't lose too much time here though, specific pages about selecting and editing will come later).

Both root and tip (the “ends”) define the bone by their respective position.

They also have a radius property, only useful for the envelope deformation method (see below).

## Vizualizace kostí

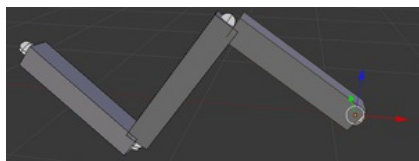
Bones can be visualized in various ways: Octahedron, Stick, B-Bone, Envelope and Wire. Custom shapes can be used, too!



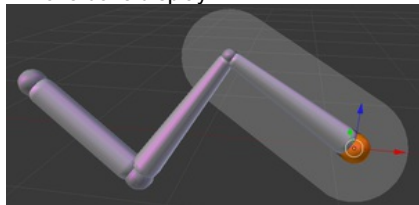
Octahedral bone display.



Stick bone display.



B-Bone bone display.



Envelope bone display.

Since armatures are made of bones, you'll find more about this when we'll talk about [Armatures Visualization](#).

Activating Axes checkmark on the Armature/Display panel, will show local axes for each bone's tip. The Y axis is always aligned along the bone, oriented from root to tip. So, this is the "roll" axis of the bones.



The Bone context.

## Vlastnosti kosti

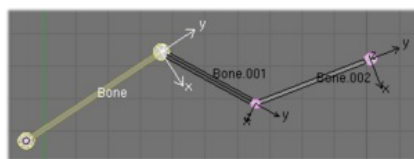
When bones are selected (hence in Edit mode and Pose mode), their properties are shown in the Bone button context of the Properties window.

This shows different panels used to control features of each selected bone, the panels change depending on which mode you're working in.

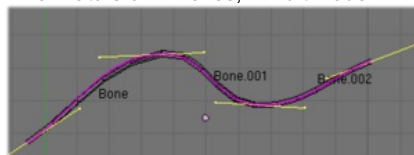
## Bones Rigidity

Even though the bones are rigid (i.e. behave as rigid sticks), they are made out of segments. Segments are small, rigid linked elements that can rotate between each other. By default, each new bone has only one segment and as such it cannot "bend" along its length. It is a rigid bone.

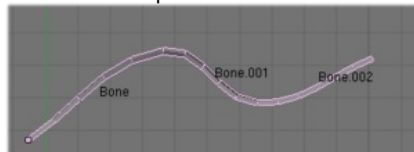
You can see these segments in Object mode and in Pose mode, and only if bones are visualized as B-bones, while in Edit mode bones are drawn as rigid sticks anyway. Note that in the special case of a single bones, you can't see these segments in Object mode, because they're aligned.



An armature of B-Bones, in Edit mode



The Bézier curve superposed to the chain, with its handles placed at bones' ends.



The same armature in Object mode

When you connect bones to form a [chain](#), Blender calculates a Bezier curve passing through all the bones' ends, and bones' segments in the chain will bend and roll to follow this invisible curve.

*You have no direct access to this curve*, you can only control it to some extent using bone properties, as explained in the [editing pages](#).

In An armature of B-Bones in Edit mode we connected 3 bones, each one made of 5 segments. These are B-bones but as you see, in Edit mode they are shown as rigid elements. Look at The same armature in Object mode: now, in Object mode, we can see how the bones' segments smoothly "blend" into each other, even for roll.

Of course, a geometry influenced by the chain is smoothly deformed accordingly to the Bezier curve! In fact, smooth bones are an easy way to replace long chains of many small rigid bones posed using IK...

However, if the chain has an influence on objects rather than geometry, the segments orientation is not taken in account (details are explained in the [skinning part](#)).

When not visualized as B-Bones, bones are always shown as rigid sticks, *even though the bones segments are still present and effective* (see [skinning to ObData](#)).

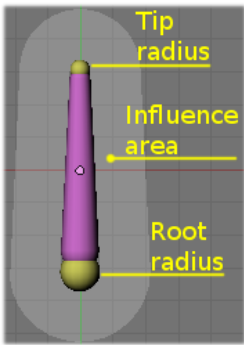
This means that even in e.g. Octahedron visualization, if some bones in a chain have several segments, they will nonetheless smoothly deform their geometry...

## Bones influence

Basically, a bone controls a geometry when vertices "follow" the bone. This is like how the muscles and skin of your finger follow your finger-bone when you move a finger.

To do this, you have to define **how much** a bone influences a certain vertex.

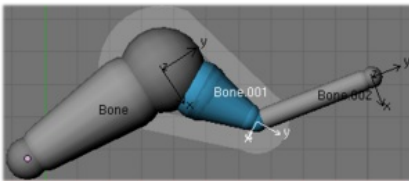
The simplest way is to have each bone affecting those parts of the geometry that are within a given range from it. This is called the *envelope technique*, because each bone can control only the geometry "enveloped" by its own influence area.



A bone in Envelope visualization, in Edit mode.

If a bone is visualized as Envelope, in Edit mode and in Pose mode you can see the area of influence, which depends on:

- the distance property
- the root's radius and the tip's radius.



Our armature in Envelope visualization, in Pose mode.

All these influence parameters are further detailed in the [skinning pages](#).

## Vlastnosti editace kostí

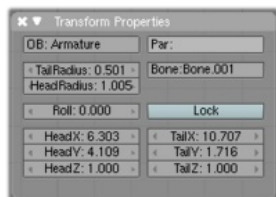
V této stránce je popsána možnost úprav a řízení většiny vlastností kostí v Blenderu.

In this page, you will learn how to edit and control most of the properties for Blender bones – For editing bones in an armature, you should read the [previous page](#) first! We will see how to [manage the bones' relationships](#), [rename them](#), etc.

## Transformování kostí

We won't detail here the various transformations of bones, nor things like axis locking, pivot points, and so on, as they are common to most object editing, and already described [here](#) (note however that some options, like snapping, do not seem to work, even though they are available...). The same goes for mirroring, as it's nearly the same as with [mesh editing](#). Just keep in mind that bones' roots and tips behave more or less like meshes' vertices, and bones themselves act like edges in a mesh.

As you know, bones can have two types of relationships: They can be parented, and in addition connected. Parented bones behave in Edit mode exactly as if they had no relations – you can grab, rotate, scale, etc. a parent bone without affecting its descendants. However, connected bones must always have parent's tips connected to child's roots, so by transforming a bone, you will affect all its connected parent/children/siblings.



The Transform Properties panel for armatures in Edit mode.

Finally, you can edit in the Transform Properties panel (N) the positions and radius of both ends of the active selected bone, as well as its [roll rotation](#).

## Radius and Scaling in Envelope Visualization

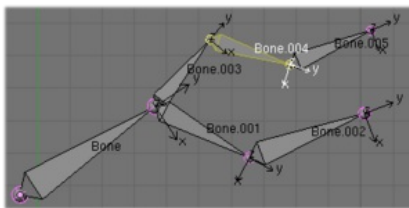
Mode: Edit mode, Envelope visualization

Hotkey: S

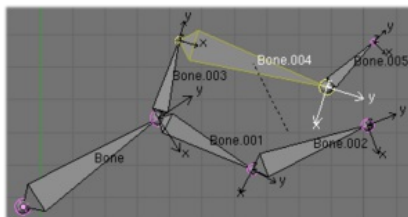
Menu: Armature » Transform » Scale

When bones are displayed using Octahedron, Stick or B-Bone visualizations, scaling will behave as expected, similar to scaling mesh objects. When bones are displayed using Envelope visualization, scaling will have a different effect: it will scale the radius of the selected bones's ends. (see: [skinning part](#)). As you control only one value (the radius), there is no axis locking here. And as usual, with connected bones, you scale at the same time the radius of the parent's tip and of the children's roots.

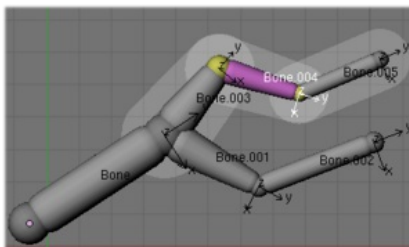
### Scaling of a bone in Octahedron and Envelope visualizations.



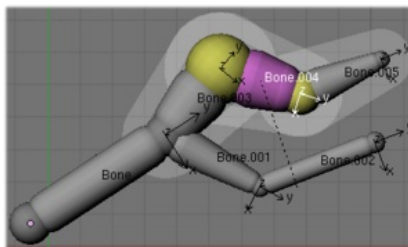
A single selected bone...



...Scaled in Octahedron visualization.



A single selected bone...



...Scaled in Envelope visualization – its length remains the same, but its ends' radius are bigger.

Note that when you resize a bone (either by directly scaling it, or by moving one of its ends), Blender automatically adjusts the end-radii of its envelope proportionally to the size of the modification. Therefore, it is advisable to place all the bones first, and only then edit these properties.

## ScaleB a Envelope - obálka

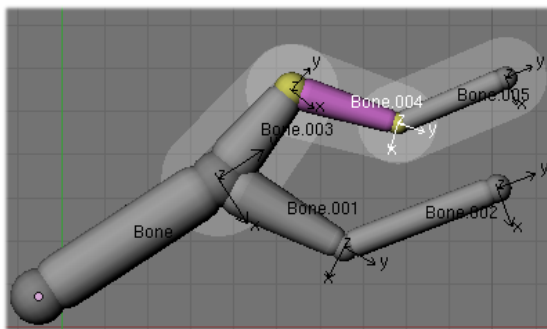
Mód: Edit mode

Klávesová zkratka: CtrlAltS

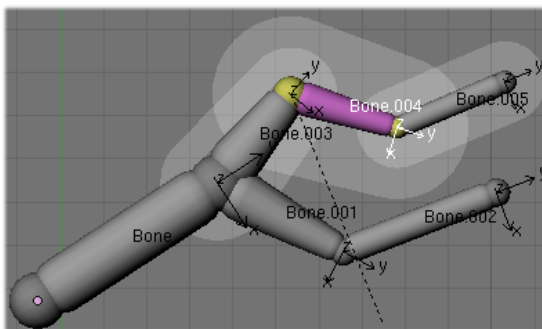
CtrlAltS aktivuje transformační nástroj specifický pro armatury. Jeho chování závisí na aktivní vizualizaci jak je vysvětleno dále:

Ve vizualizaci Envelope - obálek umožňuje editovat vliv vybrané kosti (její vlastnosti Dist, viz dokument [skinning part](#)) – jako se standardním škálováním ve vizualizaci (viz předchozí sekce). Tato vlastnost je jedno-hodnotová, a tak není třeba uvádět osu, uzamčená a pod...

### Příklady škálování obálek



Jediná kost vybraná v Envelope-obálkové vizualizaci.

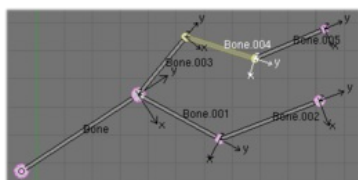


Její obálka škálovaná pomocí CtrlAltS.

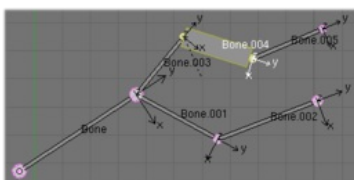
V jiných vizualizacích je umožněno upravovat "velikost kosti". Vypadá to, že jde o pouze viditelný efekt ve vizualizaci B-Bone, ale je dostupné také v Octahedron a ve Stick... Tento nástroj má trochu jiné specifické chování: Zatímco jiné transformační nástroje používají "vlastní osy" a míní se tím osy objektu, v tomto případě jde o osy vlastní dané kosti (pokud uzamknete lokální osu drojstiskem odpovídající klávesy, omezení je použito podál lokální osy samotné kosti, nikoli armatury jako celku).

**VAROVÁNÍ!** Pokud vyberete více nežli jednu kost, použití nástroje vede k pádu Blenderu!  
(což jakožto překladatel a tester musím okamžitě ověřit :-))

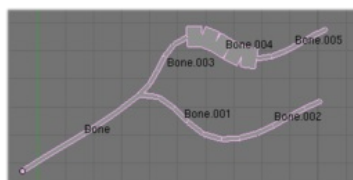
### Příklad škálování "velikosti kosti"



Samotná kost "počáteční velikost" vybraná ve vizualizaci B-Bone.



Její velikost škálovaná pomocí CtrlAltS.



Stejná armatura v objektovém módu a vizualizaci B-Bone. Kost Bone.004 zvětšena škálováním.

## Směr kosti - Bone Direction

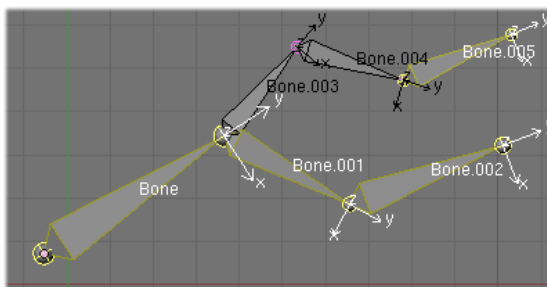
Mode: Edit mode

Hotkey: W3

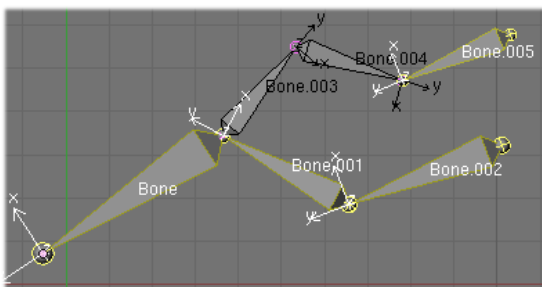
Menu: Specials » Switch Direction

Tento nástroj není dostupný z menu Armature, ale pouze z vyskakovací nabídky Specials, nebo pod zkratkou W. Umožňuje přepínat směr vybrané kosti (tj. její kořen se stane patou a naopak).

Obecně přepnutí směru kosti může způsobit přetržení řetězu jejich logických závislostí. Avšak pokud přepneme směr u celé části armatury, přepnuté kosti budou nadále spojené a zůstanou s rodičovskými vazbami - pouze v opačném pořadí. Viz *Příklad přepnutí*.



Armatura s vybranou kostí a jedním vybraným řetězem tří kostí těsně před přepnutím.



Směr vybraných kostí je otočen. Bone.005 již není nadále připojena k rodiči. Řetězek kostí však stále existuje, ale je obrácený (Nyní Bone.002 je jeho kořenem a Bone je jeho patou). Kost Bone.003 je nyní volná.

Příklad přepnutí.

## Bone Roll



Mode: Edit mode

Hotkey: CtrlR, CtrlN

Menu: Armature » Bone Roll » ...

In Edit mode, you have options dedicated to the control of the bone roll rotation (i.e. the rotation around the Y axis of the bone). Each time you add a new bone, its default roll is so that its Z axis is as perpendicular to the current 3D view as possible. And each time you transform a bone, Blender tries to determine its best roll...

But this might lead to an unclear armature, with bones rolled in all angles... nasty! To address this problem, you have three options:

- Armature » Bone Roll » Set Roll (CtrlR) will start a roll-specific rotation, which behaves like any other transform operations (i.e. move the mouse and LMB  click to validate, or type a numeric value and hit enter – or RMB  click or hit Esc to cancel everything).
- Armature » Bone Roll » Clear Roll (Z-Axis Up) (or CtrlN1 » Recalculate Bone Roll Angles » Clear Roll (Z-Axis Up)) will reset the selected bone roll so that their Z axis is as much as possible aligned with the global Z axis.
- Armature » Bone Roll » Roll to Cursor (or CtrlN2 » Recalculate Bone Roll Angles » Align Z-Axis to 3D-Cursor) will set the selected bone roll so that their Z axis is as much as possible pointed to the 3D cursor.

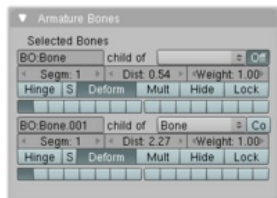
## Properties

Mode: Edit mode

Panel: Armature Bones (Editing context, F9)

Hotkey: ⇧ ShiftW, Ctrl⇧ ShiftW, AltW

Menu: Armature » Bone Settings » ...



The Armature Bones panel in Edit mode.

Most bones' properties (excepted the transform ones) are regrouped in each bone's sub-panel, in the Armature Bones panel (Editing context, F9). Let's detail them.

Note that some of them are also available in the 3D views, through the three pop-up menus Toggle Setting (⇧ ShiftW or Armature » Bone Settings » Toggle a Setting), Enable Setting (Ctrl⇧ ShiftW or Armature » Bone Settings » Enable a Setting), and Disable Setting (AltW or Armature » Bone Settings » Disable a Setting) – all three have the same entries, their respective effect should be obvious...

**BO**

The bone name field, see [below](#).

**child of**

These two settings control the bone relationship, as detailed [below](#).

**Segm**

This setting controls the number of segments that a bone has; see [below](#).

**Dist, Weight, Deform** (also ⇧ ShiftW » Deform & co), **Mult** (also ⇧ ShiftW » Mult VG & co)

These settings control how the bone influences its geometry – along with the bones' ends radius. This will be detailed in the [skinning part](#).

**Hinge** (also ⇧ ShiftW » Hinge & co), **S** (also ⇧ ShiftW » No Scale & co)

These settings affect the behavior of children bones while transforming their parent in Pose mode, so this will be detailed in the [posing part!](#)

**Hide**

This will hide the bone (same as hitting H in the 3D views, see [this page](#)).

**Lock** (also ⇧ ShiftW » Locked & co)

This will prevent all editing of the bone in Edit mode, see the [previous page](#).

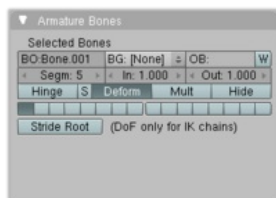
**Layers button**

These small buttons allow you to control to which bone layer this bone belongs; see [this page](#).

## Bone Rigidity Settings

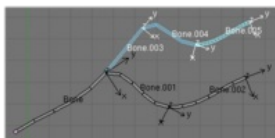
Mode: Edit and Pose modes

Panel: Armature Bones (Editing context, F9)



The Armature Bones panel in Pose mode.

Even though you have the Segm setting available in Edit mode (bones sub-panel, in the Armature Bones panel), you should switch to the Pose mode (Ctrl+ Tab) to edit these “smooth” bones’ properties – one explanation to this strange need is that in Edit mode, even in B-Bone visualization, bones are drawn as sticks, so you can’t visualize the effects of these settings.



An armature in Pose mode, B-Bone visualization:

Bone.003 has one segment,

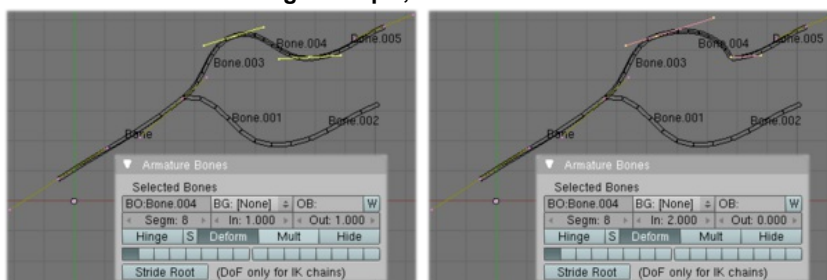
Bone.004 has four, and

Bone.005 has sixteen.

We saw in [this page](#) that bones are made of small rigid segments mapped to a “virtual” Bézier curve. The Segm numeric field allows you to set the number of segments inside a given bone – by default, it is **1**, which gives a standard rigid bone! The higher this setting (max **32**), the smoother the bone, but the heavier the pose calculations...

Each bone's ends are mapped to its “virtual” Bézier curve's “[auto](#)” handle. Therefore, you can’t control their direction, but you can change their “length” using the In and Out numeric fields, to control the “root handle” and “tip handle” of the bone, respectively. These values are proportional to the default length, which of course automatically varies depending on bone length, angle with previous/next bones in the chain, and so on.

#### Bone In/Out settings example, with a materialized Bézier curve.



Look at Bone.004: it has the default In and Out values (**1.0**). Bone.004 with In at **2.0**, and Out at **0.0**.

## Chain Editing

Mode: Edit mode

Panel: Armature Bones (Editing context, F9)

Hotkey: CtrlP, AltP

Menu: Armature » Parent » ...

You can edit the relationships between bones (and hence create/modify the chains of bones) both from the 3D views and the Buttons window. Whatever method you prefer, it's always a matter of deciding, for each bone, if it has to be parented to another one, and if so, if it should be connected to it.

To parent and/or connect bones, you can:

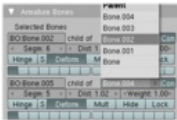
- In a 3D view, select the bone and *then* its future parent, and hit CtrlP (or Armature » Parent » Make Parent...). In the small Make Parent menu that pops up, choose Connected if you want the child to be connected to its parent, else click on Keep Offset. If you have selected more than two bones, they will all be parented to the last selected one. If you only select one already-parented bone, or all selected bones are already parented to the last selected one, your only choice is to connect them, if not already done. If you select only one non-parented bone, you'll get the Need selected bone(s) error message...

*With this method, the newly-children bones won't be scaled nor rotated – they will just be translated if you chose to connect them to their parent's tip.*

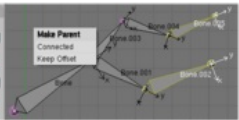
- In the Buttons window, Armature Bones panel, for each selected bone, you can select its parent in the Parent drop-down list to the upper right corner of its sub-panel. If you want them to be connected, just enable the little Con button to the right of the list.


With this method, the tip of the child bone will never be translated – so if Con is enabled, the child bone will be completely transformed by the operation.

Parenting example.

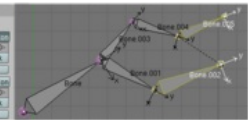


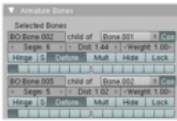
The starting armature, with Bone .005 parented and connected to Bone .004.



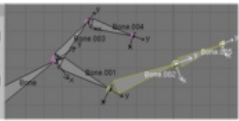



Bone .005 re-parented to Bone .002, but not connected to it (same result, using either CtrlP2 in 3D view, or the Armature Bones panel settings).



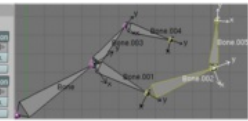


Bone .005 parented and connected to Bone .002, using CtrlP1 in 3D view.





Bone .005 parented and connected to Bone .002, using the Parent drop-down list of Bone .005 sub-panel.



To disconnect and/or free bones, you can:

- In a 3D view, select the desired bones, and hit AltP (or Armature » Parent » Clear Parent...). In the small Clear Parent menu that pops up, choose Clear Parent to completely free all selected bones, or Disconnect Bone if you just want to break their connections.
- In the Buttons window, Armature Bones panel, for each selected bone, you can select no parent in the Parent drop-down list of its sub-panel, to free it completely. If you just want to disconnect it from its parent, disable the Con button.

Note that relationships with non-selected children are never modified.

Naming Bones

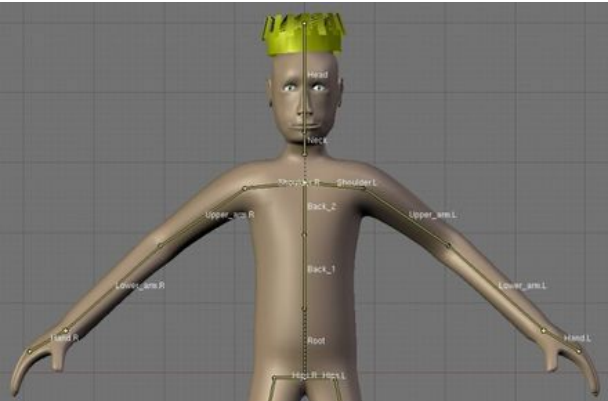
Mode: Edit mode

Panel: Armature Bones (Editing context, F9), Transform Properties (3D views, N)

You can rename your bones, either using the Bone field of the Transform Properties panel in the 3D views, for the active bone (N), or using the BO field in each bone sub-panel of the Armature Bones panel (Editing context, F9).

Blender also provides you some tools that take advantage of bones named in a left/right symmetry fashion, and others that automatically name the bones of an armature. Let's look at this in detail.

Naming Conventions



An example of left/right bone naming in a simple rig.

Naming conventions in Blender are not only useful for you in finding the right bone, but also to tell Blender when any two of them are counterparts.

In case your armature can be mirrored in half (i.e. it's bilaterally symmetrical), it's worthwhile to stick to a left/right naming convention. This will enable you to use some tools that will probably save you time and effort (like the X-Axis Mirror editing tool we saw above...).

- First you should give your bones meaningful base-names, like leg, arm, finger, back, foot, etc.
- If you have a bone that has a copy on the other side (a pair), like an arm, give it one of the following separators:
  - Left/right separators can be either the second position (L\_calfbone) or last-but-one (calfbone.R)
  - If there is a lower or upper case L, R, left or right, Blender handles the counterpart correctly. See below for a list of valid separators. Pick one and stick to it as close as possible when rigging; it will pay off. For example:

Valid Separators.		
Separator	example	
(nothing)	hand <b>Left</b>	→ hand <b>Right</b>
_ (underscore)	Hand_ <b>L</b>	→ Hand_ <b>R</b>

<code>. (point)</code>	<code>hand.l</code>	→ <code>hand.r</code>
<code>- (dash)</code>	<code>Foot-l</code>	→ <code>Foot-r</code>
<code>(space)</code>	<code>pelvis LEFT</code>	→ <code>pelvis RIGHT</code>

Note that all examples above are also valid with the left/right part placed before the name. You can only use the short `L/R` code if you use a separator (i.e. `handL/handR` won't work!).

- Before Blender handles an armature for mirroring or flipping, it first removes the number extension, if it's there (like `.001`)
- You can copy a bone named `bla.L` and flip it over using `W` » Flip Left-Right Names. Blender will name the copy `bla.L.001` and flipping the name will give you `bla.R`.

### Bone name flipping

Mode: Edit mode

Hotkey: `W4`

Menu: Armature » Flip Left & Right Names

You can flip left/right markers (see above) in selected bone names, using either Armature » Flip Left & Right Names, or Specials » Flip Left-Right Names (`W4`). This can be useful if you have constructed half of a symmetrical rig (marked for a left or right side) and duplicated and mirrored it, and want to update the names for the new side. Blender will swap text in bone names according to the above naming conventions, and remove number extensions if possible.

### Auto bone naming

Mode: Edit mode

Hotkey: `W5`, `W6`, `W7`

Menu: Armature » AutoName Left-Right, Armature » AutoName Front-Back, Armature » AutoName Top-Bottom

The three AutoName entries of the Armature and Specials (`W`) menus allows you to automatically add a suffix to all selected bones, based on the position of their root relative to the armature center and its local coordinates:

- AutoName Left-Right will add the `.L` suffix to all bones *with a positive X-coordinate root*, and the `.R` suffix to all bones *with a negative X-coordinate root*. If the root is exactly at **0.0** on the X-axis, the X-coordinate of the tip is used. If both ends are at **0.0** on the X-axis, the bone will just get a period suffix, with no L/R (as Blender cannot decide whether it is a left or right bone...).
- AutoName Front-Back will add the `.Bk` suffix to all bones *with a positive Y-coordinate root*, and the `.Fr` suffix to all bones *with a negative Y-coordinate root*. The same as with AutoName Left-Right goes for **0.0** Y-coordinate bones...
- AutoName Top-Bottom will add the `.Top` suffix to all bones *with a positive Z-coordinate root*, and the `.Bot` suffix to all bones *with a negative Z-coordinate root*. The same as with AutoName Left-Right goes for **0.0** Z-coordinate bones...

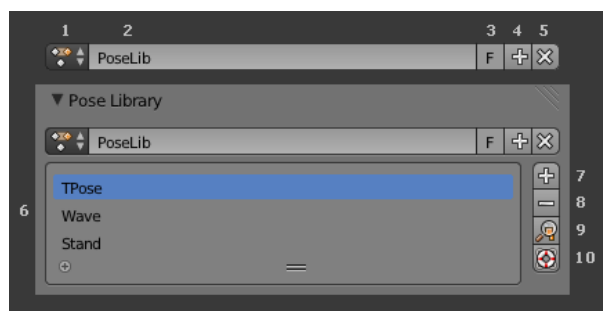
Knihovna pozic (Pose Library)

## Úvod

Panel *Knihovna pozic* se používá k uložení, aplikace a řízení různých armaturních pozic.

*Knihovny pozic (Pose Libraries)* jsou uloženy do *Aktivít (Actions)*. Knihovny obecně nejsou využité jako aktivity, ale mohou do nich nebo z nich být konvertovány.

## Panel knihovny pozic



Properties > Armature > Pose Library.

legenda k obrázku:

1. Procházení nalinkovaných *aktivit / pozičních knihoven*.
2. Název *poziční knihovny*.
3. Nastavení smyšleného (fake) uživatele.  
To umožňuje uložit knihovnu pro někoho, kdo nemá konkrétního uživatele.
4. Přidat novou *knihovnu pozic* k aktivnímu objektu.
5. Odstranit *knihovnu pozic* z aktivního objektu.
6. Výpis *Pozic* aktivní *knihovny pozic*.
7. Přidat pózu.  
Přidat novou.

Přidá novou *pózu* do aktivní *knihovny pozic* s aktuální pozicí armatury (kostry).

Přidat nový (akutální rámec - current Frame).

*Přidat nový* and *Přepsat existující* automaticky alokuje *pózu* to k rámu *Akce*.

*Přidání nového* (akutálního rámce) *přidá pózu do aktivní knihovny pozic na základě aktuálního rámce podle časového kurzoru*.

Toto není dobře podporovaná vlastnost.

Přepsat existující.

Přepsat existující *pózu* v aktivní *knihovně pozic* pro pozici aktuální armatury.

8. Odstranit existující *pózu* z aktivní *knihovny pozic*
9. Použít aktivní *pózu* na vybrané *pozicované kosti*.
10. Akce vyčištění. Zúsobí *Akcí* vhodnou pro použití *knihovny pozic*.  
Slouží k převodu *akci* na *poziční knihovnu*.  
*Pozice* je přidána do *knihovny pozic* for pro každý rámec s klíčovými snímky.

Dezinfikovat akce. Udělat *Akce* vhodný pro použití jako *Pose knihovna*.

' **Pose je přidán do Pose knihovna pro každý snímek s klíčovými snímky.**

## Úpravy

POhled 3D, poziční režim (Pose Mode).

Procházet pózy (Browse poses). CtrlL.

Přidat pózu. ⇧ ShiftL.

Přejmenovat pózu. ⇧ ShiftCtrlL.

Vymazat pózu. AltL.

## Omezovače

Omezovače patří mezi funkce, které se vztahují k *transformačním vlastnostem* (pozice, rotace a měřítko) mezi objekty navzájem. Omezovače jsou jakýmsi protipólem [modifikátorů](#), které naopak reagují pouze s *daty* samotného objektu (tj. sítě, křivky atd.).

Všechny omezovače mají společné základní [rozhraní](#) podobně tak, jako tomu mají modifikátory.

## Použití omezovačů

Even though constraints might be very useful in static scenes (as they can help to automatically position/rotate/scale objects), they were first designed for animation, as they allow you to limit/control the freedom of an object, either in absolute (i.e. in global space), or relatively to other objects.

Also note that constraints internally work using 4x4 transformation matrices only. When you use settings for specific rotation or scaling constraining, this information is being derived from the matrix only, not from settings in a Bone or Object. Especially for combining rotations with non-uniform or negative scaling this can lead to unpredictable behavior.

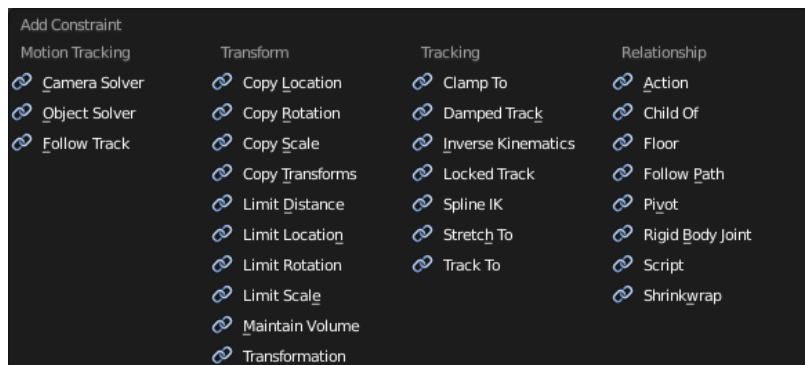
## Omezování kostí

Finally, there is a great rigging feature in Blender: in Pose mode, each bone of an armature behaves a bit like a standard object, and, as such, can be constrained. Most constraints work well with both objects and bones, but there are a few exceptions which are noted in the relevant constraints pages.

To learn more:

- Read about using constraints in object animation in the [Animation chapter](#)
- Read about using constraints in rigging in the [Armatures](#)

## Dostupné omezovače



The Constraint Menu

There are several types of constraints. We can classify them into four families:

- Motion Tracking
- [Transform](#)
- [Tracking](#)
- [Relationship](#)

There are constraints that works with their *owner* object and others that need a second object (the *target*) to work, sometimes of a specific type (e.g. a curve). In this case targeted constraints are shown as a dark blue dashed line drawn in the 3D view between the owner and target objects.

## Sledování pohybu

Camera Solver  
Object Solver  
Follow Track

## Transformační omezovače

These constraints directly control/limit the transform properties of its owner, either absolutely or relatively in terms of its target properties.

<a href="#">Copy Location</a>	Copies the location of the target (with an optional offset) to the owner, so that both move together.
<a href="#">Copy Rotation</a>	Copies the rotation of the target (with an optional offset) to the owner, so that both rotate together.
<a href="#">Copy Scale</a>	Copies the scale of the target (with an optional offset) to the owner, so that both scale together.
<a href="#">Copy Transforms</a>	Copies the transforms of the target to the owner, so that both transform together.
<a href="#">Limit Distance</a>	Limits the position of the owner, so that it is nearer/further/exactly at the specified distance from the target.
<a href="#">Limit Location</a>	Limits the owner's location inside a given range.
<a href="#">Limit Rotation</a>	Limits the owner's rotation inside a given range.
<a href="#">Limit Scale</a>	Limits the owner's scale inside a given range.

<a href="#">Transformation</a>	Uses a property of the target (location, rotation or scale), to control a property (the same or a different one) of the owner.
<a href="#">Maintain Volume</a>	Maintains the volume of a bone or an object.

### Dráhové omezovače

These constraints try, in various ways, to adjust their owner's properties so that it "points at" or "follows" the target.

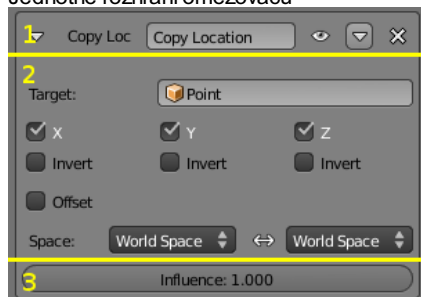
<a href="#">Clamp To</a>	Clamps the owner to a given curve target.
<a href="#">Damped Track</a>	
<a href="#">Inverse Kinematics</a>	Bones only. Creates a chain of bones controlled by the target, using inverse kinematics.
<a href="#">Locked Track</a>	The owner is tracked to the given target, but with a given axis' orientation locked.
<a href="#">Spline IK</a>	
<a href="#">Stretch To</a>	Stretch the owner to the given target.
<a href="#">Track To</a>	The owner is tracked to the given target.

### Vztahové omezovače

These are "misc" constraints.

<a href="#">Action</a>	The owner executes an action, controlled by the target (driver).
<a href="#">Child Of</a>	Allows a selective application of the effects of parenting to another object.
<a href="#">Floor</a>	Uses the target's position (and optionally rotation) to define a "wall" or "floor" that the owner won't be able to cross.
<a href="#">Follow Path</a>	The owner moves along the curve target.
<a href="#">Pivot</a>	
<a href="#">Rigid Body Joint</a>	Creates a rigid joint (like a hinge...) between the owner and the "target" (child object).
<a href="#">Script</a>	Uses a Python script as constraint.
<a href="#">Shrinkwrap</a>	Limits the location of the owner at <i>the surface</i> (among other options) of the target.

### Jednotné rozhraní omezovačů



Tři části uživatelského rozhraní

As with [modifiers](#), an object (or bone, see the [rigging chapter](#) for details) can use several constraints at once. Hence, these constraints are organized in a stack which controls their order of evaluation (from top to bottom).

All constraints share a common basic interface, packed up in a sort of sub-panel, that is split into three parts:

1. The header, gathering most common settings.
2. The constraint's specific settings.
3. The influence and animation controls (the Rigid Body Joint constraints have no influence setting).

### Záhlaví omezovače



Záhlaví omezovače

The header of a constraint “sub-panel” is the same for all. From left to right, you have:

A small arrow

This control allows you to show/hide the constraint's settings.

The constraint type

This is just static text showing you what this constraint is...

The name field

Here you can give your constraint a more meaningful name than the default one.

This control has another *important* purpose: it turns red when the constraint is not functional (as in *A constraint header*). As most constraints need a second “target” object to work (see below), when just added, they are in “red state”, as Blender cannot guess which object or bone to use as target. This can also happen when you choose an invalid set of settings, e.g. with a [Track To constraint](#) of which the To and Up vectors are both set to the same axis.

As noted above, constraints in “red state” are ignored during the stack evaluation.

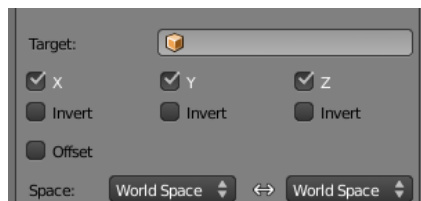
The “up”/“down” buttons

As seen above, these allow you to move a constraint up/down in the stack.

The “X” control

As seen above, this will delete (remove from the stack) the constraint.

### Nastavení omezovače



The central part of a constraint's subpanel contains the constraint's settings, the target, and constraint space

The constraints settings area is of course specific to each constraint type. However, there are two points that are common to many constraints, so we will detail them here.

#### Cíl (target)

Most constraints need another “target” object or bone to “guide” them. You select which by selecting its name in the Target field. Except for a few cases, you can use any type of object (camera, mesh, empty...); its object origin will be the target point.

When you type in the OB field a mesh or lattice name, a second Vertex Group field appears just below. If you leave it empty, the mesh or lattice will be used as a standard object target. But if you enter in this Vertex Group field the name of one of the mesh's or lattice's vertex groups, then the constraint will use the median point of this vertex group as target.

Similarly, if you type in the OB field an armature name, a second Bone field appears just below. If you enter in it the name of one of the armature's bones, then the constraint will use this bone's *root* as target. In some constraints, when you use a bone as target, another Head/Tail numeric field will also appear, that allows you to select where along the bone the target point will lay, from root (**0.0**) to tip (**1.0**) (remember that currently, in Blender UI, bones' roots are called "heads", and bones' tips, "tails"...).

## = Prostor (Space) omezovače

For many constraints you can choose in which space it is evaluated/applied. In the Space drop-down lists, the right side one is the space that the owner is evaluated in (Owner Space). When such a constraint uses a target, you can also choose in which space the target is evaluated (Target Space). The Target Space drop-down list is on the left side. Both lists have the same options, depending on whether the element (owner or target) is a regular object, or a bone:

### Local Space

The object's properties are evaluated in its own local space, i.e. based on its rest position (without taking into account its parents' transformations in its chain, or its armature object's transformation).

### Local With Parent (bones only)

The bone properties are evaluated in its own local space, *including* the transformations due to a possible parent relationship (i.e. due to the chain's transformations above the bone).

### Pose Space (bones only)

The bone properties are evaluated in the armature object local space (i.e. independently from the armature transformations in Object mode). Hence, if the armature object has null transformations, Pose Space will have the same effect as World Space...

### Local (Without Parent) Space (objects only)

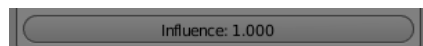
The object properties are evaluated in its own local space, *without* the transformations due to a possible parent relationship.

### World Space (default setting)

Here the object's or bone's properties are evaluated in the global coordinate system. This is the easiest to understand and most natural behavior, as it always uses the "visual" transform properties (i.e. as you see them in the 3D views).

Understanding the Constraint Space effects is not really easy (unless you are a geometry genius...). The best thing to do is to experiment with their different combinations, using e.g. two empties (as they materialize clearly their axes), and a Copy Rotation constraint (as rotations are the most demonstrative transformations, to visualize the various spaces specificities...).

## Vliv (Influence)

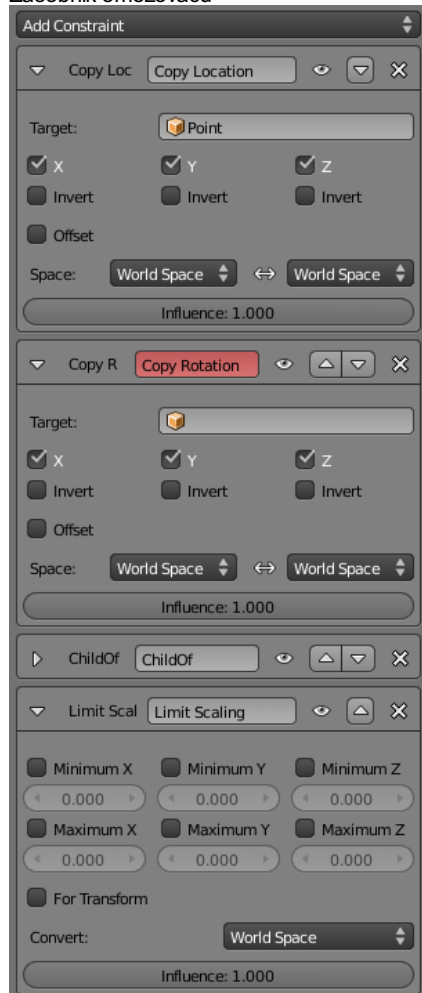


Influence

At the bottom of nearly all constraints, you have the Influence slider, which controls the influence of the constraint on its owner. As you might expect, **0.0** means that the constraint has no effect, and **1.0** means that the constraint has full effect. Using in-between values, you can have several constraints all working together on the same owner's properties. Note that if a constraint has a full influence on a given property, all other constraints above in the stack working on that same property will have no effect at all.

But the best thing with influence is that you can animate it with an Fcurve – see [the constraints page of the animation chapter](#) for more details about this.

## Zásobník omezovačů



Příklad zásobníka omezovačů

Omezovače jsou zpracovávány zhora dolů jak je patrné v panelu Constraints.

1. Jako první v uvedeném příkladu (*A constraint stack example*) je zpracováno umístění osvětlení (kopírována na objekt vlastníka).
2. Omezení kopie rotace je ignorováno (červený název, viz níže).
3. Je vyhodnoceno další dceřinné omezení Child Of, což je redukce.
4. Nakonec je velikost krychle omezena hodnotou měřítka Limit Scale.

A tak je velikost krychle řízena cílem jako první Child Of omezení, a to uvnitř limitních hodnot daných dalším omezovačem Limit Scale constraint... Pořadí je velice důležité!

Jednotlivé omezovače je možné v zásobníku posouvat nahoru a dolů pomocí malých šipek v pravé části lišty každého omezovače.

## Přidání/odstranění omezovače

Přidání omezovače je možné pomocí panelu Constraints, a to kliknutím na tlačítko Add Constraint. V menu je vypsán seznam dostupných omezovačů pro právě aktivní objekt (nebo kost - bone v mopozičním (pose) modelovacím režimu (ve kterém jsou také omezení zobrazeny v menu omezovače kostí). Nový omezovač je přidán *vždy* nakonec (tedy na vrchol) zásobníku.

You can also, in a 3D view, either:

- Select only the future owner, hit Ctrl+Shift+C, and in the Add Constraint to New Empty Object menu that pops up, select the constraint you want to add. If the chosen constraint needs it, a new Empty object will be automatically added as target, positioned at the owner's center, and with null rotation.
- Select first the future target, and then the future owner, hit Ctrl+Shift+C, and in the Add Constraint to Active Object (or Add Constraint to Active Bone) menu that pops up, select the constraint you want to add. If the chosen constraint needs it, the other selected object/bone will be used as target.

Note that these pop-up menus do not display all the available constraints.

To remove a constraint, click on the "X" button of the header of the constraint you want to delete, in the Constraints panel. You can also remove all constraints from the selected object(s), using the Object » Constraints » Clear Object Constraints (or Pose » Constraints » Clear Pose Constraints... or hit Ctrl+Alt+C).

Omezovač kopírování umístění

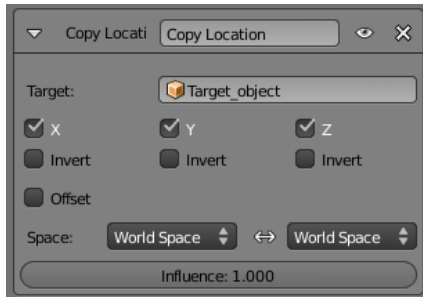
## Popis

Omezovač Kopie umístění (Copy Location) umožňuje jeho vlastníkovvi mít stejné umístění jako jeho cíl.



Note that if you use such a constraint on a *connected* bone, it will have no effect, as it is the parent's tip which controls the position of your owner bone's root.

## Volby



Panel kopie umístění

### Target

This constraint uses one target, and is not functional (red state) when it has none.

#### Bone

If Target is an Armature, a new field is displayed offering the optional choice to set an individual bone as Target.

#### Head/Tail

If a Bone is set as Target, a new field is displayed offering the optional choice of where along this bone the target point lies.

#### Vertex Group

If Target is a Mesh, a new field is displayed offering the optional choice to set a Vertex Group as target.

### X, Y, Z

Tyto tlačítka určují, podle kterých směrů (os) omezovač působí. Implicitně podle všech tří směrů.

#### Invert

The Invert buttons invert their respective preceding coordinates.

### Offset

When enabled, this control allows the owner to be translated (using its current transform properties), relative to its target's position.

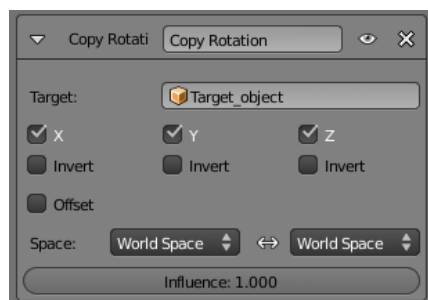
### Space

This constraint allows you to choose in which space to evaluate its owner's and target's transform properties.

Omezovač kopie rotace

Omezovač kopie rotace potlačuje možnost svévolné otáčení objektu a nutí jej otáčet se podle cíle.

## Volby



Panel Copy Rotation

### Cíl - Target

Tento omezovač využívá jeden cíl je nefunkční (červeně), pokud není cíl zvolen.

#### Kost - Bone

Pokud Target je Armaturou, nově zobrazené pole nabízí možnost volitelného nastavení individuální kosti jako cíle.

#### Hlava/ocas - Head/Tail

Pokud je kost Bone nastavena jako cíl Target, je nové pole zobrazeno s nabídkou volby nastavení ve kterém ležé cílový bod poldél směru kosti.

#### Vertex Group

Pokud je Target sítí Mesh, nové pole se zobrazí s nabídkou volitelné monosti vybrat skupinu Vertex Group jako cíl.

### X, Y, Z

Tyto tlačítka určují, které osy budou součástí omezení. Implicitně jsou nastaveny všechny tři.

#### Invert

Tlačítko inverze Invert obrací jejich hodnoty rotace.

### Posuv - Offset

Pokud je nastaveno, tato kontrola umožňuje jeho nositeli být rotován (za použití jeho běžných transformačních vlastností), relativně k orientaci jeho cíle.

### Prostor - Space

Toto omezení dává modeláři možnost zvolit ve kterém prostoru se bude omezovač vyhodnocovat.

akce (actions) == akce == ;when? animující objekty a vlastnosti v Blender akce záznam a obsahuje data

[File:K animation actions](#)

[data3.png](#)

actions.

;so? když vy animovat objekt skrze měnící se jeho umístění s klíčové snímky animace je uložen k akce ;každý vlastnost má kanál který to je nahané k pro příklad je nahané k kanál ray umístění každý kanál má f-křivka s sata z klíčové snímky

[File:K animation actions](#)

[keyframes.png](#)

graph editor každý kanál

má f-křivka který jsou

hladký linky klíčové snímky

jsou řídící points.

;actions : záznam a obsahuje animace data ;groups : jsou skupiny z kanály ;channels : záznam vlastnosti ;f-curves : vytvořit klíčové snímky hladký s interpolovaní ;keyframes : jsou použito k sata hodnoty z vlastnosti

[File:K animation actions](#)

[create.png](#)

actions.

když vy první animovat objekt skrze přidávat klíčové snímky Blender vytváří *action*? k záznam data nový akce může bude vytvořen a vy může použít a změna akce v výkres - akce editor nebo nla? editor když vy jsou použití násobně akce pro stejný objekt stisk **f**? tlačítko pro každý *unlinked? action?* toto bude dát akce *fake? user?* a bude zamezit Blender z smazání akce k smazat předchozí a vytvoření nový prázdný akce odlinkovat akce skrze skisknutím **x**? tlačítko pak přidat některý nový klíčové snímky k vytvoření nový akce Blender nebude uložit *unlinked?* akce k ponechat předchozí a vytvoření nový prázdný akce stisk **f**? pak odlinkovat skrze skisknutím **x**? tlačítko pak přidat některý nový klíčové snímky k vytvoření nový akce k ponechat předchozí a duplikovaný akce stisk **f**? pak stisk **+**? tlačítko ;you? může zapéci akce skrze vybírá objekt skisknutím prostor sloupek pak psaní 'bake? action'? pak skisknutím 'bake? action'? operátor toto bude zapéci animace k nový akce pro více info viz [Python 2.69.1 zapéci action](#)

== kanály ==

[File:Doc animation driver](#)

[fcurve.png](#)

graph editor: ? kanál f-

curve.

kanály použít čas k řídící hodnoty z vlastnosti použití f-křivka s klíčové snímky kanál čte čas rámec a sady hodnota z vlastnost tak z toto příklad když čas rámec je 2.0 vlastnost bude bude 0.5 nejvíce nastavení pro f-křivka jsou najít v graf editor

== základní animace == ;here? jsou některý cesty k animovat objekty skrze přidávat klíčové snímky k kanály ;these? metody může bude použito na rozdíl objekty jako armatura kosti v póza režim === vložit klíčové snímky ===


[File:Doc kia cube00.png](#)

add klíčové snímky k

cube.

[File:Doc kia cube01.png](#)

keyframes v timeline.

časová osa editor přesun zelený rámec kurzor k 0 skrze skisknutím nebo držící LMB  ;with? síť objekt vybraný v objekt režim a rámec sata v časová osa v 3d pohled nástroje stisk 'insert? klíčové snímky tlačítko nebo stisk **I** v 3d pohled k přinášet nahoru 'insert? klíčové snímky menu ;select? 'location'? toto bude záznam umístění z síť objekt na rámec časová osa kurzor je sata ;the? první čas vy dělat toto toto bude přidat akce k objekt s umístění skupina ray y z kanály a toto bude přidat klíčové snímky k každý kanál ;set? časová osa kurzor k rozdíl rámec přesun síť objekt pak vložit jiný klíčové snímky ;this? bude nyní přidat jiný sata z klíčové snímky k umístění kanály ;the? klíčové snímky jsou zobrazen v žlutá na časová osa přesun časová osa kurzor k test animace === klíčování sady ===

[File:Doc kia cube02.png](#)

timeline klíčování sets.

;keying? sady jsou sata z klíčové snímky kanály v jeden oni jsou dělají k záznam násobně vlastnosti v stejný čas tam jsou některý built? v klíčování sady 'locrotscale'? a také custom? klíčování sady ;use? klíčování sata menu v časová osa editor k vybrat klíčování sata nyní když vy vložit klíčové snímky Blender bude přidat klíčové snímky k vše kanály v vybraný klíčování sata ;see? [timeline](#) pro více info ;see?

[keying sets](#) pro více info === auto klíčové snímky ===


[File:Doc kia cube03.png](#)

timeline auto keyframe.

;auto? klíčové snímky je červený záznam tlačítko ;auto? klíčové snímky přidává klíčové snímky automaticky k sata rámec po nastavení nový hodnota pro vlastnost ;see? [timeline](#) pro více info === vlastnosti ===

[File:Doc kia cube04.png](#)

keyframe properties.

;keyframes? může bude použito k animovat mnoho z vlastnosti v Blender ;to? přidat klíčové snímky k vlastnost na ui **RMB**  vlastnost pak vybrat vložit samotný klíčové snímky nebo vložit klíčové snímky ;insert? klíčové snímky bude přidat klíčové snímky k sata z vlastnosti === klávesnice === ;the? klávesnice může také bude použito k přidat klíčové snímky :!/? vložit klíčové snímky k sata rámec :Alt+i:? odstranit klíčové snímky z sata rámec :Alt+shift+i:? odstranit vše klíčové snímky z ui vlastnost == viz také == .? == links? == .? '?

ovladače (drivers) == drivers? ==

[File:Doc animation driver](#)

[fcurve.png](#)

graph editor:? driver?

example.

drivers? může použít vlastnosti numbers? transformace a skripty k řídicí hodnoty z vlastnosti použití f-křivka driver? čte hodnota z driver? hodnota a sady hodnoty z vybraný vlastnost to bylo přidán k tak z toto příklad když driver? hodnota je 2.0 vlastnost bude bude 0.5 driver? hodnota je určuje skrze driver? variables? nebo scripted? expression? nejvíce nastavení pro drivers? [f-curves](#) jsou najít v [graph editor](#)

== přidávat drivers? == ;to? přidat driver? k vlastnost najít vlastnost vy chce k přidat driver? k [Template:Shortcut/mouse](#) na vlastnost a vybrat jeden z následující options:? ;add? drivers :this? bude přidat drivers? k sata z properties? related? k vybraný jeden :for? příklad to bude přidat drivers? k ray y a z pro rotace ;add? samotný driver :this? bude přidat samotný driver? k vybraný vlastnost

[File:Doc add driver2.png](#)

add samotný driver.

== drivers? panel ==

[File:Doc animation panel](#)

[drivers.png](#)

graph editor:? drivers?:

drivers? panel.

;this? panel je pro nastavení nahoru driver? variables? nebo scripted? expression? který určuje hodnota z driver? hodnota ===== driver? nastavení ===== ;update? dependencies?: toto bude force? update? pro driver? hodnota dependencies? ;remove? driver?: removes? driver? z objekt ;type?: typ z výpočty k použití na driver? variables? ;average? value?: užívají average? hodnota ;sum? values?: užívají sum? ;scripted? expression?: užívají scripted? expression? viz expr? ;this? option? does? ne použít variables? ale oni může bude přidán k scripted? expression? ;minimum? value?: užívají lowest? hodnota ;maximum? value?: užívají highest? hodnota ;expr?: scripted? expression? :here? vy může přidat reálný numbers? math? operátor math? funkce Python vlastnosti driver? funkce :see? driver? expression? níže pro některý příklady ;show? debug? info?: ukazuje driver? hodnota ;the? current? hodnota z variables? nebo scripted? expression? ;add? variable?: přidává nový driver? variable?

[File:Doc animation driver](#)

[single property.png](#)

setup z samotný property.

[File:Doc animation driver](#)

[transform channel2.png](#)

transform kanál setup.

[File:Doc animation driver](#)

[distance.png](#)

distance setup.

===== driver? variables? ===== ;name?: jméno k použít pro scripted? expressions/functions? ;: ne spaces? nebo dots? jsou allowed? a musí start s letter? ;variable? type?: typ z variable? k použít ;{{literal|single? vlastnost použít hodnota z některý ma? vlastnost ;:for? příklad ambient? stínovaný color? z materiály ;:first? vybrat typ z id-block? pak id? z id-block? pak kopie a paste? ma? vlastnost (ctrl+v) ;:id-type?: id-block? typ příklad key? obrázek objekt materiály ;:id?: id? z id-block? typ příklad materiály ;:ma? path?: ma? id? jméno z property? příklad 'ambient'? z materiály stínovaný ;transform? channel?: použít jeden z transform? kanály z objekt nebo bone? ;:id?: id? z objekt příklad krychle armatura kamera ;:bone?: id? z armatura bone? příklad bone? bone.002? arma.r? ;:;? toto option? je pro armatura ;:type?: příklad ray umístění ray rotace ray scale? ;:space?: world? prostor transformace prostor local? prostor ;{{literal|rotational? rozdíl použít rotace rozdíl mezi dva objekty nebo kosti ;distance?: použít distance? mezi dva objekty nebo kosti ;value?: ukazuje hodnota z variable?

== příklady == ===== driver? expression? ===== zde jsou některý příklady použití scripted? expression? expr? k sata driver? hodnota

[File:Doc animation driver](#)

[object rotation.png](#)

object rotation.

===== orbit? point? ===== zde dva drivers? mít been? přidán k krychle ray umístění a y umístění scripted? expressions? jsou být použito k sata objekt umístění ;x? umístění expr?: **0+(sin(frame/8)\*4)?** ;: (frame/8):? je current? rámec z animace divided? skrze 8 k slow? orbit? down? ;: (sin(?)\*4):? toto returns? sine? z (frame/8)? pak multiplies? skrze 4 pro bigger? circle? ;: 0+:? je použito k řídicí ray umístění offset? z orbit? ;y? umístění expr?: **0+(cos(frame/8)\*4)?** ;: (frame/8):? je current? rámec z animace divided? skrze 8 k slow? orbit? down? ;: (cos(?)\*4):? toto returns? cosine? z (frame/8)? pak multiplies? skrze 4 pro bigger? circle? ;: 0+:? je použito k řídicí y umístění offset? z orbit? **frame?** je stejný jako bpy.context.scene.frame\_current? ===== driver? namespace? ===== ;there? je seznam z built? v driver? funkce a vlastnosti ;these? může bude zobrazen skrze Python console? <source? lang=python>>>? bpy.app.driver\_namespace['? \_\_builtins\_\_? \_\_doc\_\_? \_\_loader\_\_? \_\_name\_\_? \_\_package\_\_? \_\_acos? \_\_acosh? \_\_asin? \_\_asinh? \_\_atan? \_\_atan2? \_\_atanh? \_\_bpy? \_\_ceil? \_\_copysign? \_\_cos? \_\_cosh? ..</source>? ;this? script? bude přidat funkce k driver? namespace? který může pak bude použito v expression? **driverfunc(frame)?** <source? lang=python>import? bpy? def? driverfunc(val):? return? val? \* val? #? return? val? squared? bpy.app.driver\_namespace['driverfunc'] = driverfunc? #? přidat funkce k driver\_namespace</source>? == tvar key? driver? == ;this? příklad je tvar key? driver? driver? bylo přidán k tvar key? hodnota

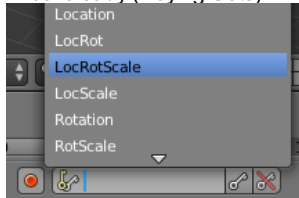
[shape key? driver? kliknutí k enlarge.](#)

;this? příklad užívají armatura bone? 'b'? z rotace k řídicí hodnota z tvar key? ;the? bone? rotace režim je sata k xyz euler? ;the? driver? f-křivka je mapovány jako tak :bone? z rotace 0.0(0.0): tvar key? hodnota 0.0 :bone? z rotace -2.09(-120.0): tvar key? hodnota 1.0

;this? kind? z driver? může také bude setup? s variable? typ rotace rozdíl ;see? [shape keys](#) pro více info == viz také == [\\*animation](#)

[\\*graph? editor? \\*f-curves? \\*extending? Blender s python](#) == links? == [\\*python](#) a jeho [documentation](#) [\\*functions.wolfram.com](#) ?

## Klíčové sady (Keying Sets)



Časová osa klíčových sad.

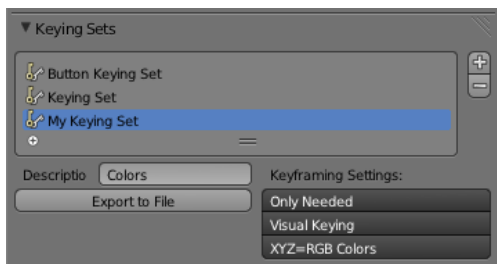
Klíčové sady jsou kolekce vlastností. Jsou využity v klíčování více vlastností ve stejném okamžiku, obvykle stiskem klávesy I v pohledu 3D.

Některé sady jsou připravené, nazývané *Absolutní sady klíčů*.

Pro použití klíčových sad zvolte *Active Keying Set* v [Timeline Header](#), nebo v panelu *Keying Set Panel*, anebo stiskněte Ctrl+Alt+Shift v pohledu 3D.

## Panel klíčových sad

Tímto panelem se dá přidat, vybrat, spravovat absolutní *Absolute Keying Sets*.



Properties > Scene > Keying Set Panel.

### Jméno - Keying Set Name

Aktivní klíčová sada je zvýrazněna modře, LMB X2 pro přejmenování.

+

Přidá novou (prázdnou) klíčovou sadu do aktivní scény.

-

Odstraní aktivní klíčovou sadu.

### Vlastnosti klíčových sad - properties

#### Description - popis

Stručný popis významu klíčové sady.

#### Export do souboru

Export klíčové sady do skriptu Pythonu *File.py*.

Pro přečtení klíčové sady ze souboru *File.py* otevřte a susťte *File.py* v textovém editoru.

#### Nastavení klíčování

Tyto volby řídí všechny vlastnosti klíčové sady.

Poznámka: stejné nastavení je v uživatelských volbách *User Preferences*, které přebíjí tato nastavení pokud je tato volba pololena.

#### Pouze potřebné - Only Needed

Vkládá pouze klíčování tam, kde je nutné v relevanci se křivkami typu F-curve.

#### Vizuální klíčování

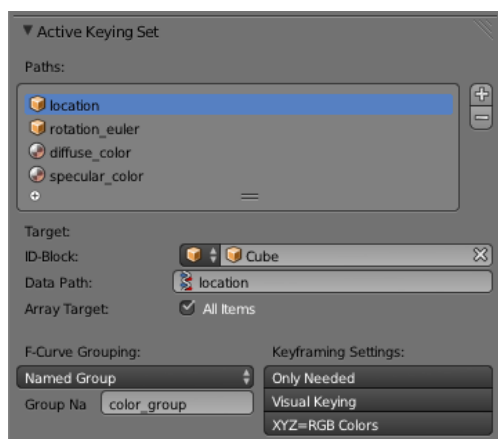
Vkládá klíčování na základě vizuální transformace.

#### XYZ=RGB Colors

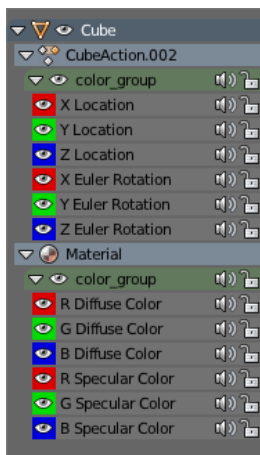
For new F-Curves, set the colors to RGB for the property set, Location XYZ for example.

## Panel aktivní klíčové sady

Tento panel je určen pro přidání vlastností aktivní klíčové sady.



Properties > Scene > Active Keying Set Panel.



Properties > Graph Editor  
> Channels, Named  
Group.

### Paths - cesty

A collection of *Paths* each with a *Data Path* to a property to add to the active Keying Set. The active *Path* is highlighted in blue.

+

Add new empty path to active Keying Set.

-

Remove active path from the active Keying Set.

### Active Path properties

#### ID-Block

Set the *ID-Type* + *Object ID Data Path* for the property.

#### Data Path

Set the rest of the *Data Path* for the property.

#### Array Target

Use *All Items* from the *Data Path* or select the array index for a specific property.

#### F-Curve Grouping

This controls what *Group* to add the *Channels* to.  
*Keying Set Name*, *None*, *Named Group*.

#### Keyframing Settings

These options control individual properties in the Keying Set.

##### Only Needed

Only insert keyframes where they're needed in the relevant F-Curves.

##### Visual Keying


Insert keyframes based on the visual transformation.

##### XYZ=RGB Colors

For new F-Curves, set the colors to RGB for the property set, Location XYZ for example.

## Přidávání vlastností

Some ways to add properties to keying sets.

RMB  the property in the *User Interface*, then select *Add Single to Keying Set* or *Add All to Keying Set*. This will add the properties to the active keying set, or to a new keying set if none exist.

Hover the mouse over the properties, then press K, to add *Add All to Keying Set*.

## Viz také

- [Timeline Header - V Keyframe Control](#)

## Časová osa (Timeline)

Okno časové osy je značeno ikonou hodin. Jeeho umístění je přednastavené ve spodní části pracovní plochy.


Okno časové osy není ve skutečnosti "editor", ale víceméně informační okno s omezenými možnostmi úprav..

Zde je vidět přehled, v jakém časovém okamžiku se nachází celá scéna, kolik rámečků je za sekundu, kde se nachází klíčové snímky právě aktivního objektu, jaký je počáteční a koncový rámeček animace, a tak podobně...

Stějn jako na videozařízení VTR jsou zde ovládací prvky přehrát animaci, pozastavit, změnit aktuální rámeček, rolovat v čase mezi rámečky (VTR je zkratkou pro Video Tape Recorder, klasický "videopřehrávač" z dob na přelomu tisíciletí).

## Prvky časové osy

### Aktuální snímek

The current frame is materialized by the thick green vertical line (the so-called "time cursor", in *Timeline window*, it is at frame **100**). You can move it by clicking LMB  anywhere in the window, and you can even scroll forth and back the animation by clicking and dragging with this same mouse button. The actual frame number (or second value) is drawn near the pointer when you click or drag the time cursor – and obviously, it is always in the "current frame" numeric field of the [header](#).

### Klíčové snímky

Some keyframes of the active object (or active [Doc:2.5/Manual/Animation/Keyframes](#), etc.) are materialized by vertical colored lines at the frame they occur. I think only three different lpo types are drawn this way:

- **žlutá**

Non-action *Object* keys (location, rotation, etc.)

- **oranžová**

*Material* keys (diffuse/specular/mirror colors, etc.)

- **temně azurová**

keys forming [actions](#) (always pose of armature's bones, but also possibly objects' and shapes' lpos)

Note

Some lpo type keys aren't materialized at all (e.g. *Texture* or *Constraint* ones).

In [\(Timeline window\)](#), we have two "object" keyframes (frames **10** and **150**), one "material" keyframe (frame **40**), and no "action" keyframe.

### Značky (Markers)

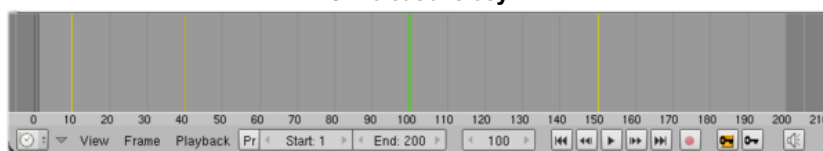
[Markers](#) are materialized as small triangles, with their name near them.

Barevné označení znamenají:

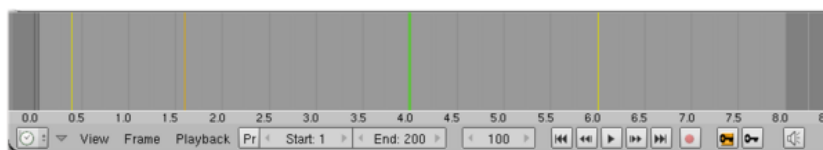
- **bílá linka line, černý text:** nevybrané značky
- **žlutá linka, bílý text:** vybrané značky

## Nastavení pohledu

Okno časové osy.






Zobrazení snímků.



Zobrazení sekund.

Toto okno je jednorozměrné - podél horizontální osy představuje pouze čas ve scéně. Podobně, jakou jsou snímky základní jednotkou času v Blenderu, časová osa zobrazuje ve výchozím nastavení čísla snímků.

Animační rozsah je zvýrazněn světle šedou oblastí (v okně časové osy od snímku **1** do snímku **200**).

Toto okno má vlastnosti jakékoli jiné oblasti v Blenderu: je možné jej posouvat MMB , klik-a-táhnout (pouze doleva/doprava), zvětšovat a zmenšovat Ctrl MMB  nebo kolečkem Wheel , atp.

**Menu pohled**

**Úpravy**

**Menu rámečků**

**Přehrát**

**Menu přehrávání**

**Header Controls**

## Klíčové snímky (Keyframes)

Klíčové snímky jsou základem animace. Definují hodnotu dat v určitém čase.

Je zřejmé, že pokud je něco "animované", tak se mění v čase. Animace objektů v Blenderu znamená změnu mnoha parametrů scény. Od polohy jednotlivých předmětů, až po jejich geometrické a fyzikální vlastnosti, tj. například rychlost, tvar, barva, osvětlení.

Základní jednotkou pro animaci v Blenderu je "rámeček", nebo okénko (frame). Ten má význam zobrazení jednoho obrazu po určitou dobu projekce, kdy samozřejmě závisí na frekvenci snímků (frame rate), tedy v případě 25 snímků za sekundu trvá jeden rámeček 40 milisekund.

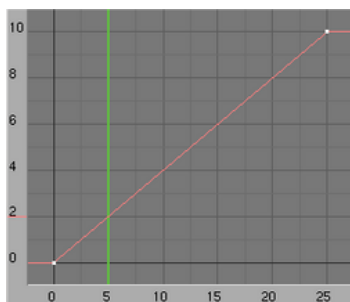
Tak, jako je animace tvořena postupnými změnami parametrů, obvykle nejsou vlastnosti objektů měněny ručně styem "rámeček po rámečku" jako tomu bývá u tvorby animovaných filmů pomocí fotokamery a figurek. Výhody jsou zřejmé:

- zabralo by to celý věk animátora!
- bylo by velice obtížné vystihnout přesné pohybové charakteristiky objektu včase, plynulé, nebo měnící se pohyby a pod.

Proto se animace v Blenderu dá poněkud přesněji vystihnout pojmem **interpolace**, což matematické vkládání a nalezení přibližné hodnoty funkce v daném intervalu.

Tedy Blender je založen na jednoduché myšlence definovat několik "kontrolních bodů" nazývaných "**klíčové snímky**", mezi nimiž se na časové ose vyskytuje více "okének".

Mezi těmito klíčovými snímky jsou hodnoty parametrů scény dopočítávány Blenderem. Tak má animátor značně zjednodušenou práci, neboť pouze definuje klíčové snímky a typ průběhu hodnot mezi nimi. Tento dokument [křivky](#) může být upravován pomocí [Editor grafů](#).



Příklad interpolace

Jako příklad mějme:

- kontrolní bod s hodnotou **0** na rámečku **0**,
- další hodnotu **10** na rámečku **25**,
- lineární (přímkovou) interpolaci,

potom v rámečku **5** dostáváme hodnotu **2**.

Úplně stejná situace je pro rámečky mezi: s pouhými dvěma body je možné získat hladkou změnu od **0** do **10** během **25 rámečků**. Obvykle pokud požadujete na snímku **15** mít hodnotu **9**, musíte přidat další kontrolní bod nebo klíčový snímek...

## Vytváření klíčových snímků


### v pohledu 3D

Klíčové snímky jsou nastavovány horkou klávesou I. Nabídkové menu dává následující možnosti:

- Location - umístění
- Rotation - rotace
- Scale - měřítko
- LocRot - místo+rot
- LocScale - místo+měř
- LocRotScale - místo+rot+měř
- RotScale - rot+měř
- Visual Location - vizuální umístění
- Visual Rotation - vizuální rotace
- Visual Scale - vizuální měřítko
- Delta Location - změna umístění
- Delta Rotation - změna rotace
- Delta Scale - změna měřítka

### v panelu vlastností

Je možné přidat klíčový snímek pro většinu modifikovatelných atributů. Toto je možné podržením daného parametru a stiskem požadavku vzniku klíče horkou klávesou:

- I. Pole je zažluceno a indikuje, že rámeček byl nastaven.
- Alt+i. Vymazání aktuálního rámečku.
- Alt+shift+i. Vymazání všech rámečků v tomto kanálu (tj. týkající se tohoto parametru).
- RMB  na atributu a vyberte vložení Insert Keyframes, nebo Insert Single Keyframe pro zpracování vektorových atributů.

## v animačním editoru

Je možné přidat klíčový snímek v kanálech, kde již existují klíčové snímky mnohými způsoby popsány na těchto stránkách:

- [Timeline - časová osa](#)
- [Graph Editor - editor grafů](#)
- [Dope Sheet - úkolník](#)

Pozn. Anglický pojem "dope-sheet" má český ekvivalent "list s pokyny". Pravděpodobně nejvhodnější a nejsrozumitelnější pojem pro překlad je "úkolník".

## Úpravy klíčových snímků

Existuje mnoho způsobů úprav klíčových snímků. Vesměs jsou popsány v uvedených dokumentech výše.

## Úvod do fyzikální simulace

Tato kapitola popisuje různé efekty často používané při simulaci skutečných fyzikálních jevů pro zobrazení objektů jako jsou:

- kouř
- déšť
- mlha
- látka
- voda
- rosol

[Částicové systémy](#) mohou být využity k simulaci mnoha objektů jako jsou například vlasy, tráva, kouř, atp.

[Vlasy](#) jsou jednou z podmnožin částicových systémů a mohou být užity pro vláknovité objekty jako tráva, kožešina a podobně.

[Měkká tělesa - Soft Bodies](#) jsou užitečná všude tam, kde dochází k jevům deformace, svázání, reakce na vnější síly jako gravitace, nebo kde dochází ke kolizím s dalšími objekty. Mohou být použity pro kůži, gumu a dokonce oblečení, přestože pro 'oblékání' obsahuje Blender velice specifickou [simulaci oblečení](#).

[Pevná tělesa - Rigid Bodies](#) mohou simulovat dynamiku objektů, které jsou skutečně pevné.

[Kapaliny - Fluids](#) zahrnují simulaci kapalin a plynů včetně například [kouře](#).

[Sílová pole - Force Fields](#) mají vliv na další okolní chování simulací.

## Gravitace

Gravitace patří mezi globální nastavení která má stejný účinek na všechny fyzikální systémy ve scéně. Obecně má nastavenou hodnotu -9.810 směrem v ose Z, což je skutečná hodnota gravitace na zemském povrchu (fyzikální značka gravitace je  $Literallg$  a jednotka  $[m \times s^{-2}]$ ). Snížením této hodnoty je možné docílit pohybu v menším tíhovém poli (například Měsíce). Samosebou hodnota 0 reprezentuje stav beztlže.

Tíhové zrychlení dosahuje po celé **Zemi** téměř stejné hodnoty. Na severním, či jižním pólu má hodnotu  $g = 9,832 \text{ m} \times \text{s}^{-2}$ , na rovníku  $g = 9,780 \text{ m} \times \text{s}^{-2}$ . Pro precizní výpočty Praha, nebo Boston má hodnotu 9,81373. Pro renderování scén z **Měsíce** je vhodné počítat s jeho tíhovým zrychlením, 6 krát menším nežli zemském ( $1,622 \text{ m} \times \text{s}^{-2}$ ). Populární a pravděpodobně neosídlená planeta **Mars** má  $g = 3,69$ .

**Změnu nastavení** velikosti gravitace je možné změnit v záložce Tíhové pole - Field Weights.

Pro nezletilé

Pro Maxe a školáky, kteří ještě neměli ve škole fyziku je vhodné podotknout, že čím vyšší  $g$ , tím větší gravitace. Nulové  $g$  (stav beztlže) zvládne jen Sandra Bullocková ve filmu Gravity. Přetížení 6g stíhací letci při prudkých otáčkách strojů atp...)

## Silová pole

Force Fields offer a way to add extra movement to dynamic systems. [Particles](#), [Soft Bodies](#), [Rigid Bodies](#) and [Cloth objects](#) can all be affected by forces fields. Force Fields automatically affect everything. To remove a simulation or particle system from their influence, simply turn down the influence of that type of Force Field in its Field Weights panel.

- Všechny typy objektů a částic mohou tvořit silová pole, ?but only curve object can ?bear Curve Guides pole.
- Silová pole mohou být generovány částicemi. Viz [Fyzika částic](#)
- Objekty potřebují sdílet alespoň jednu společnou vrstvu pro to, aby byl jev patrný.

Je možné omezit efekt na částice na skupinu objektů. Více na stránce [Fyzika částic](#).

## Vytvoření silového pole

Mód: Object Mode

Panel: Object kontext → Physics-fyzika podkontext → Fields-pole

Klávesová zkratka: F7

Pro vytvoření jednoduchého silového pole vyberte Add » Force Field a vyberte požadovaný typ pole. Tato metoda vytvoří objekt Empty (prázdné) s připojeným silovým polem.

Pro vytvoření silového pole z existujícího objektu musíte vybrat objekt a změnit jeho fyzikální podkontext. Vyberte typ pole v menu Fields.

Pole mají mnoho nastavení a běžně se dají vysvětlit na sférickém (Spherical) poli.

### Note

After changing the fields (Fields panel) or deflection (Collision panel) settings, you have to recalculate the particle, softbody or cloth system (Free Cache), this is not done automatically. You can clear the cache for all selected objects with CtrlB → Free cache selected.

Particles react to all kind of Force Fields, Soft Bodies only to Spherical/Wind/Vortex (they react on Harmonic fields but not in a useful way).

## Common Field Settings

Most Fields have the same settings, even though they act very differently. Settings unique to a field type are described below. Curve Guide and Texture Fields have very different options.

### Shape

The field is either a Point, with omnidirectional influence, or a Plane, constant in the XY-plane, changes only in Z direction.

### Strength

The strength of the field effect. This can be positive or negative to change the direction that the force operates in. A force field's strength is scaled with the force object's scale, allowing you to scale up and down scene, keeping the same effects.

### Flow

Convert effector force into air flow velocity.

### Noise

Adds noise to the strength of the force.

### Seed

Changes the seed of the random noise.

### Effect Point

You can toggle the field's effect on particle Location and Rotation

### Collision Absorption

Force gets absorbed by collision objects.

## Falloff

Here you can specify the shape of the force field (if the Fall-off Power is greater than 0).

### Sphere

Falloff is uniform in all directions, as in a sphere.

### Tube

Fall off results in a tube shaped force field.

The Field's Radial falloff can be adjusted, as well as the Minimum and Maximum distances of the field.

### Cone

Fall off results in a cone shaped force field. Additional options are the same as those of Tube options.

### Z Direction

Fall-off can be set to apply only in the direction of the positive Z Axis, negative Z Axis, or both.

### Power (Power)

How the power of the force field changes with the distance from the force field. If  $r$  is the distance from the center of the object, the force changes with  $1/r^{\text{Power}}$ . A Fall-off of 2 changes the force field with  $1/r^2$ , which is the falloff of gravitational pull.

Max Distance - maximální vzdálenost

Makes the force field only take effect within a specified maximum radius (shown by an additional circle around the object).

Min Distance

The distance from the object center, up to where the force field is effective with full strength. If you have a Fall-off of 0 this parameter does nothing, because the field is effective with full strength up to Max Dist (or the infinity). Shown by an additional circle around the object.

## Typy polí

### silové

The Force field is the simplest of the fields. It gives a constant force towards (positive strength) or away from (negative strength) the object's center. Newtonian particles are attracted to a field with negative strength, and are blown away from a field with positive strength.

For [Boids](#) a field with positive strength can be used as a *Goal*, a field with negative strength can be used as *Predator*. Whether Boids seek or fly goals/predators depends on the Physics settings of the Boids.



Image 2b:  
Spherical field  
indicator.

### vítr

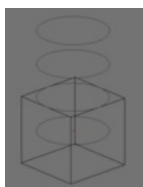


Image 3a:  
Wind field  
indicator.

Wind gives a constant force in a single direction, along the force object's local Z axis. The strength of the force is visualized by the spacing of the circles shown.

### vírové

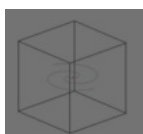


Image 3b:  
Vortex field  
indicator.

Vortex fields give a spiraling force that twists the direction of points around the force object's local Z axis. This can be useful for making a swirling sink, or tornado, or kinks in particle hair.

### magnetické

This field depends on the speed of the particles. It simulates the force of magnetism on magnetized objects.

### harmonické

The source of the force field is the zero point of a harmonic oscillator (spring, pendulum). If you set the Damping parameter to 1, the movement is stopped in the moment the object is reached. This force field is really special if you assign it to particles.

Rest Length

Controls the rest length of the harmonic force.

Multiple Springs

Causes every point to be affected by multiple springs.

Normally every particle of the field system influences every particle of the target system. Not with Harmonic! Here every target particle is assigned to a field particle. So particles will move to the place of other particles, thus forming shapes. [Tutorial: Particles forming Shapes](#).

### náboj

It is similar to spherical field except it changes behavior (attract/repulse) based on the effected particles charge field (negative/positive), like real particles with a charge. This mean this field has only effect on particles that have also a Charge field (else, they have no "charge", and hence are unaffected)!

## Lennard-Jonesovo

Toto pole je velice krátkodosahové a dosahuje jak odpudivé pro menší vzdálenosti, tak odpudivých charakteristik pro vzdálenosti větší. Toto pole má snahu držet částice v rovnovážné vzdálenosti (equilibrium distance) jedna od druhé. Obvykle se tyto síly kombinují se síly dané rozdílným nábojem.

Particles can have for example both a charge and a Lennard-Jones potential - which is probably something for the nuclear physicists amongst us.

## Pole textur

You can use a texture force field to create an arbitrarily complicated force field, which force in the 3 directions is color coded. Red is coding for the x-axis, green for the y-axis and blue for the z-axis (like the color of the coordinate axes in the 3D window). A value of 0.5 means no force, a value larger than 0.5 acceleration in negative axis direction (like -Z), a value smaller than 0.5 acceleration in positive axis direction (like +Z).

### Texture mode

This sets the way a force vector is derived from the texture.

#### RGB

Uses the color components directly as the force vector components in the color encoded directions. You need an RGB texture for this, e.g. an image or a colorband. So a Blend texture without a colorband would not suffice.

#### Gradient

Calculates the force vector as the 3d-gradient of the intensity (grayscale) of the texture. The gradient vector always points to the direction of increasing brightness.

#### Curl

Calculates the force vector from the curl of the 3d-rgb texture (rotation of rgb vectors). This also works only with a color texture. It can be used for example to create a nice looking turbulence force with a color clouds texture with perlin noise.

### Nabla

It is the offset used to calculate the partial derivatives needed for Gradient and Curl texture modes.

### Use Object Coordinates

Uses the emitter object coordinates (and rotation & scale) as the texture space the particles use. Allows for moving force fields, that have their coordinates bound to the location coordinates of an object.

### Root Texture Coordinates

This is useful for hair as it uses the texture force calculated for the particle root position for all parts of the hair strand.

### 2D

The 2D button disregards the particles z-coordinate and only uses particles x&y as the texture coordinates.

Remember that only procedural texture are truly 3D.

## Příklady

- A single colored texture 0.5/0.0/0.5 creates a force in the direction of the positive y-axis, e.g. hair is orientated to the y-axis.
- A blend texture with colorband can be used to created a force "plane". E.g. on the left side 0.5/0.5/0.5, on the right side 1.0/0.5/0.5 you have a force plane perpendicular to XY (i.e. parallel to Z). If you use an object for the coordinates, you can use the object to push particles around.
- An animated wood texture can be used to create a wave like motion.

## Curve Guide

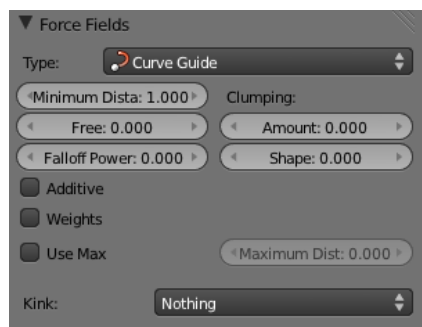


Image 4a: A Curve Guide field.

Curve objects can be the source of a Curve Guide field. You can guide particles along a certain path, they don't affect Softbodies. A typical scenario would be to move a red blood cell inside a vein, or to animate the particle flow in a motor. You can use Curve Guides also to shape certain hair strands - though this may no longer be used as often now because we have the [Particle Mode](#). Since you can animate curves as Softbody or any other usual way, you may build very complex animations while keeping great control and keeping the simulation time to a minimum.

The option Curve Follow does not work for particles. Instead you have to set Angular Velocity (in the Physics panel of the Particle sub-context) to Spin and leave the rotation constant (i.e. don't turn on Dynamic).

Curve Guides affect all particles on the same layer, independently from their distance to the curve. If you have several guides in a layer, their fields add up to each other (the way you may have learned it in your physics course). But you can limit their influence radius:

#### Minimum Distance

The distance from the curve, up to where the force field is effective with full strength. If you have a Fall-off of 0 this parameter does nothing, because the field is effective with full strength up to MaxDist (or the infinity). MinDist is shown with a circle at the endpoints of the curve in the 3D window.

#### Free

Fraction of particle life time, that is not used for the curve.

#### Fall-off

This setting governs the strength of the guide between MinDist and MaxDist. A Fall-off of 1 means a linear progression.

A particle follows a Curve Guide during it's lifetime, the velocity depends from it's lifetime and the length of the path.

#### Additive

If you use Additive, the speed of the particles is also evaluated depending on the Fall-off.

#### Weights

Use Curve weights to influence the particle influence along the curve.

#### Maximum Distance/Use Max

The maximum influence radius. Shown by an additional circle around the curve object.

The other settings govern the form of the force field along the curve.

#### Clumping Amount

The particles come together at the end of the curve (1) or they drift apart (-1).

#### Shape

Defines the form in which the particles come together. +0.99: the particles meet at the end of the curve. 0: linear progression along the curve. -0.99: the particles meet at the beginning of the curve.

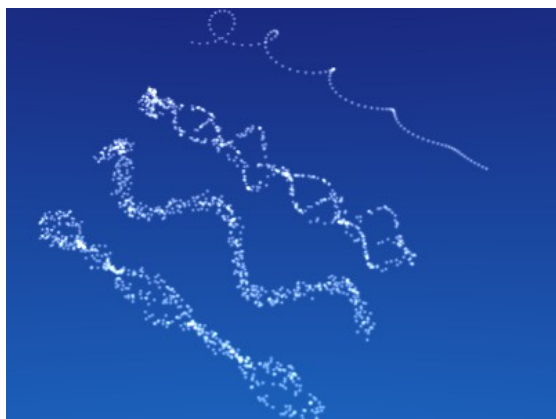


Image 4b: Kink options of a curve guide. From left to right: Radial, Wave, Braid, Roll.

[Animation](#)

With the drop down box Kink, you can vary the form of the force field:

#### Curl

The radius of the influence depends on the distance of the curve to the emitter.

#### Radial

A three dimensional, standing wave.

#### Wave

A two dimensional, standing wave.

#### Braid

Braid.

#### Roll

A one dimensional, standing wave.

It is not so easy to describe the resulting shapes, I hope it's shown clearly enough in (*Image 4b*).

#### Frequency

The frequency of the offset.

#### Shape

Adjust the offset to the beginning/end.

#### Amplitude

The Amplitude of the offset.

## Boid

Název Boid pochází z počítačového programu vyvinutého Craigem Reynoldsem v roce 1986. Program simuluje chování letu ptáků. Jeho práce na toto téma byla publikována v roce 1987 ve sborníku konference ACM SIGGRAPH. Stejně jako u většiny simulací

umělého života je Boids příkladem důležitého chování. Složitost Boids vyplývá ze vzájemného ovlivňování jednotlivých prvků (dále boids v tomto případě ) podchycené sadou jednoduchých pravidel. Pravidla uplatňovaná v nejjednodušším prostředí Boids jsou následující:

- oddělení, separace - prvky se chovají tak, aby nevytěšňovaly sousední
- zarovnání - směr pohybu se v průměru blíží k prvním letcům
- soudržnost, koheze - pohyb prvků směřuje tak, že průměrné místní těžiště místních prvků se nemění.

Samozřejmě mohou být přidána k chování "Boids" daleko složitější další pravidla.

## Turbulence

Create a random turbulence effect with a 3d noise.

Size

Indicates the scale of the noise.

Global

Makes the size and strength of the noise relative to the world, instead of the object it is attached to.

## Tažení, brždění

Drag is a force that works to resist particle motion by slowing it down.

Linear

Drag component proportional to velocity.

Quadratic

Drag component proportional to the square of the velocity.

## Odkazy

- [Wind & Deflector force update 2.48](#)
- [Particle options and guides \(v2.40\)](#)

## Kolize

Se síťovými obkety mohou kolidovat částice ([Particles](#)), [Soft Bodies](#) a objekty oblečení ([Cloth objects](#)). Objekty typu Boid ([Boids](#)) se kolizím vyhýbají.

- Objekty potřebují sdílet alespoň jednu společnou vrstvu.
- Je možné omezit vliv částic na skupinu objektů (v panelu tíhové pole [Field Weights](#)).
- *Deflection* for softbody objects is difficult, they often penetrate the colliding objects.
- Částice vlasů (Hair) ignorují deflexi objektů (ale je možné je animovat jako softbodies které již deflexi respektují).

Je možné změnit nastavení deflexe pro objekt. Tím se přepočítají částice, softbody nebo systém oblečení. Je třeba uvolnit keš ((Free Cache), což neprobíhá automaticky. Je možné keš vymazat pro všechny vybrané objekty pomocí zkratky CtrlB → Free cache selected.

Mode: Object Mode

Panel: Object context → Physics sub-context → Collision

## Nastavení

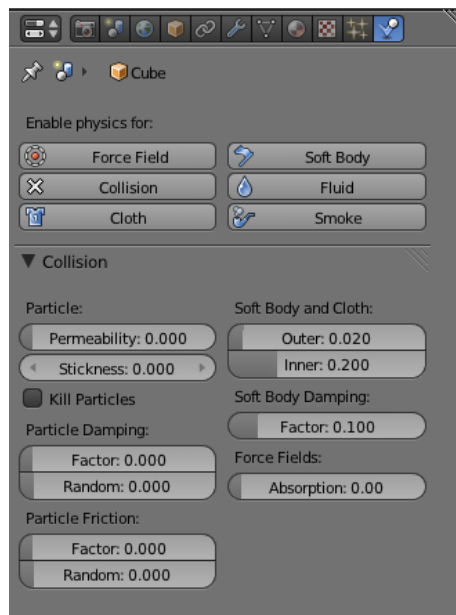


Image 1: Panel Collision v podkontextu Physics.

### Permeability

Fraction of particles passing through the mesh. Can be animated with Object lpos, Perm channel.

### Stickiness

How much particles stick to the object.

### Kill Particles

Deletes Particles upon impact.

### Damping Factor

Damping during a collision (independent of the velocity of the particles).

### Random damping

Random variation of damping.

### Friction Factor

Friction during movements along the surface.

### Random friction

Random variation of friction.

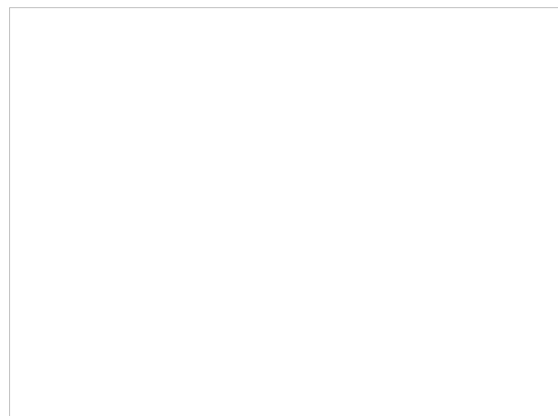


Image 1b: A softbody vertex colliding with a plane.

## Interakce Soft Body a Oblečení

### Outer

Size of the outer collision zone.

### Inner

Size of the inner collision zone (padding distance).

Outside and inside is defined by the face normal, depicted as blue arrow in (*Image 1b*).

### Damping Factor

Damping during a collision.

*Softbody* collisions are difficult to get perfect. If one of the objects move too fast, the soft body will penetrate the mesh. See also the section about [Soft Bodies](#).

## Interakce se silovými poli

### Absorption

A deflector can also deflect effectors. You can specify some collision/deflector objects which deflect a specific portion of the effector force using the Absorption value. 100% absorption results in no force getting through the collision/deflector object at all.

If you have 3 collision object behind each other with e.g. 10%, 43% and 3%, the absorption ends up at around 50% ( $100 \times (1 - 0.1) \times (1 - 0.43) \times (1 - 0.03)$ ).

## Příklady



Image 2: Deflected Particles.

Here is a Meta object, duplivered to a particle system emitting downwards, and deflected by a mesh cube:

## Tipy

- Make sure that the normals of the mesh surface are facing towards the particles/points for correct deflection.
- Hair particles react directly to force fields, so if you use a force field with a short range you don't need necessarily collision.
- Hair particles avoid their emitting mesh if you edit them in Particle mode. So you can at least model the hair with collision.

## Částice (Particles)

Mezi "částice" zahrnujeme velkou spoustu prvků emitovaných ze síťoviny objektů, obvykle v tisících. Každá částice může být světelným bodem nebo sítí a může být dynamicky propojená s ostatními. Na částice může působit mnoho různých vlivů a sil, mohou mít vlastní životnost. Dynamické částice mohou představovat oheň, kouř, mlhu a další materiály jako je prach nebo samotná kouzla.

[Hair](#) type particles are a subset of regular particles. Systém vlasů (hair particles) tvořený prvky, které mohou také představovat vlasy, srst, trávu, či štětiny.

Vidíte částice jako modifikátor Particle, ale všechna nastavení se provádí v záložce Particle tab.

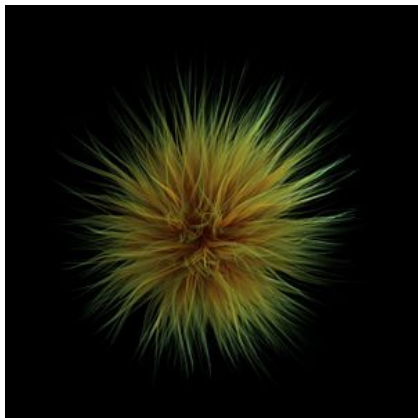


Image 1: Some fur made from particles  
([Blend file](#)).

Částice obecně vytékají ze své rodné sítě do prostoru. Jejich pohyb může být ovlivněn mnohými skutečnostmi, včetně:

- Počáteční rychlost.
- Pohyb emiteru - vyslače (vrchol, stěna, objekt).
- Pohyb v závislosti na "gravitaci" nebo "odporu vzduchu".
- Vliv silových polí, jako je vítr, víry nebo pohyb podél křivky.
- Interakce s jinými předměty, jako jsou kolize.
- Částečně inteligentní chování členů stáda (stádo, školy, ...), které reagují s ostatními členy svého stáda, a zároveň se snaží dosáhnout cíle, nebo vyhnout dravcům.
- Hladký pohyb s pomocí fyziky měkkých částí (pouze částicové systémy typu Hair).
- Nebo dokonce pouze manuální transformace.

## [Lattices](#).

Particles may be rendered as:

- [Halos](#) (for Flames, Smoke, Clouds).
- Meshes which in turn may be animated (e.g. fish, bees, ...). In these cases, each particle "carries" another object.
- [Strands](#) (for [Hair](#), [Fur](#), [Grass](#)); the complete way of a particle will be shown as a strand. These strands can be manipulated in the 3D window (combing, adding, cutting, moving, etc).

Every object may carry many particle systems. Each particle system may contain up to 100.000 particles. Certain particle types (Hair and Keyed) may have up to 10.000 children for each particle (children move and emit more or less like their respective parents). The size of your memory and your patience are your practical boundaries.

## Incompatibility with Prior Versions

There are many differences between the "old" particle system that was used up to and including version 2.45, and the "new" particle system. There are many things possible now that could not be done with the old system. The new system is incompatible to the old system, though Blender tries to convert old particle systems, which works only to some extent. The old system is most like the new Emitter system (keep reading to find out what that is). If you are using an old version of Blender 2.45 and previous, [click here to access the old documentation](#).

## Postup - Workflow

The process for working with standard particles is:

1. Create the mesh which will emit the particles.
2. Create one or more Particle Systems to emit from the mesh. Many times, multiple particle systems interact or merge with each other to achieve the overall desired effect.
3. Tailor each Particle System's settings to achieve the desired effect.
4. Animate the base mesh and other particle meshes involved in the scene.
5. Define and shape the path and flow of the particles.
6. For [Hair](#) particle systems: Sculpt the emitter's flow (cut the hair to length and comb it for example).
7. Make final render and do physics simulation(s), and tweak as needed.

## Vytvoření částicového systému

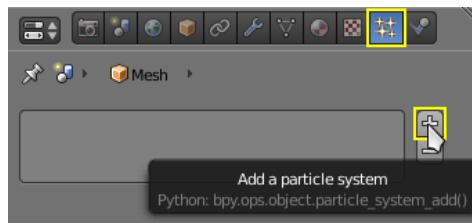


Image 2: Adding a particle system.

To add a new particle system to an object, go to the Particles tab of the object Settings editor and click the small + button. An object can have many Particle Systems.

Each particle system has separate settings attached to it. These settings can be shared among different particle systems, so one doesn't have to copy every setting manually and can use the same effect on multiple objects. Using the Random property they can be randomized to look slightly different, even when using the same settings.

### Typy částicových systémů

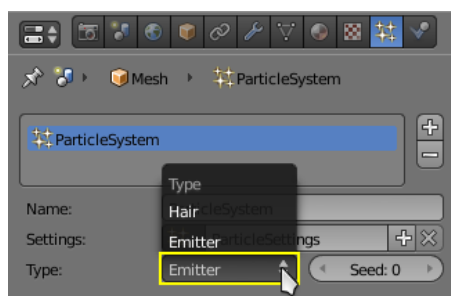


Image 3: Particle system types.

After you have created a particle system, the Property window fills with many panels and buttons. But don't panic! There are two different types of particle systems, and you can change between these two with the Type drop-down list:

#### Emitter

This parallels the old system to the greatest extent. In such a system, particles are emitted from the selected object from the Start frame to the End frame and have a certain lifespan.

#### Hair

This system type is rendered as strands and has some very special properties: it may be edited in the 3D window in realtime and you can also animate the strands with [Cloth Simulation](#).

The settings in the Particle System panel are partially different for each system type. For example, in *Image 3* they are shown for only system type Emitter.

### Společné volby

Each system has the same basic sets of controls, but options within those sets vary based on the system employed. These sets of controls are:

<a href="#">Emission</a>	Settings for the initial distribution of particles on the emitter and the way they are born into the scene.
<a href="#">Cache</a>	In order to increase realtime response and avoid unnecessary recalculation of particles, the particle data can be cached in memory or stored on disk.
<a href="#">Velocity</a>	Initial speed of particles.
<a href="#">Rotation</a>	Rotational behavior of particles.
<a href="#">Physics</a>	How the movement of the particles behaves.
<a href="#">Render</a>	Rendering options.
<a href="#">Display</a>	Realtime display in the 3D View.
<a href="#">Children</a>	Control the creation of additional child particles.
<a href="#">Field Weights</a>	Factors for external forces.
<a href="#">Force Field</a>	Makes particles force fields.
<a href="#">Settings</a>	
<a href="#">Vertex Groups</a>	Influencing various settings with vertex groups.

## Odkazy

- [Tutorials](#)
- [Physics Caching and Baking](#)
- [Particle Rewrite Documentation](#)
- [Thoughts about the particle rewrite code](#)
- [Static Particle Fur Library](#)

Motion Tracking (sledování pohybu kamery)

## Úvod

Motion tracking je novou technikou dostupnou v Blenderu. Stále je ve vývoji a zatím podporuje jen základní operace pro sledování pohybu kamery. Nicméně již je připraven pro použití.

## Začínáme

Motion tracking je dostupný v aktuálním SVN trunku a bude zahrnut v Blenderu verze 2.61. Je dostupný pro všechny platformy a může být použit "out-of-box".

Zde je stručný popis již v Blenderu dostupných nástrojů pro motion tracking.

### Dohlížení na 2D tracking

Neexistuje žádný obecný algoritmus, který by mohl být použit pro všechny možné záběry, charakteristické body a jejich pohyb. Takový algoritmus by mohl být vytvořen, nicméně by byl velmi pomalý a stejně by nemusel fungovat bezchybně. Jedinou možností jak provádět 2D tracking je, manuálně volit vhodný algoritmus a jeho nastavení. Současné výchozí nastavení by mělo pěkně fungovat pro většinu záběrů, které nejsou moc rozmazaný a kde charakteristické body příliš nedeformuje perspektivní zobrazení, atd.

Vylepšený 2D tracking již je v TODO seznamu, ale aktuálně nepatří mezi priority. Pokud si nejste jisti algoritmy, jejich nastavením a nechcete číst tento dokument, můžete si pohrát s nastavením a najít tak to, které vám přinese požadovaný výsledek.

### Manuální kalibrace objektivu pomocí grease pencil a/nebo mřížky

Všechny kamery, vlivem chyb objektivů, zaznamenávají zkreslený obraz, s čímž nic udělat nemůžeme. Pro zjištění přesného pohybu kamery potřebujeme znát přesnou hodnotu ohniskové vzdálenosti a "sílu" zkreslení.

V současnosti můžeme ohniskovou vzdálenost zjistit z nastavení kamery nebo z EXIFu -- Blender si tuto hodnotu neumí zjistit sám. Je tu však několik nástrojů, které mohou pomoci najít přibližné hodnoty a kompenzovat zkreslení. K dispozici je také plně manuální nástroj, který umožňuje použití sítě deformované modelem zkreslení, která definuje přímky v záběru. Můžete také použít grease pencil - stačí nakreslit linku pomocí "poly line brush", která by měla být přímo na záběru, a upravit hodnoty zkreslení tak, aby korespondovala s linií v záznamu.

Jedinou možností pro přesnější kalibraci vaší kamery je použití nástroje "grid calibration" z knihovny OpenCV. OpenCV používá stejný model zkreslení, takže by to neměl být problém.

### Řešení pohybu kamery

Přesto že není žádný rozdíl v detekci pohybu kamery a pohybu objektů, alespoň z matematického úhlu pohledu, je zatím k dispozici jen pohyb kamery. Toto řešení má zatím své hranice, mezi které patří chybějící podpora pohybu kamery na stativu nebo pohyb dominantní roviny (kde je všechny pohyb vstažen k jedné rovině). V budoucnu by tyto nedostatky měly být odstraněny.

### Základní nástroje pro orientaci a stabilizaci scény

Po detekci pohybu je potřeba nastavit správnou orientaci reálné scény v 3D prostoru, kvůli pohodlnější práci. K dispozici je nástroj pro definici "podlahy", středu scény a x/y-ové osy, které umožňují scénu orientovat.

Někdy je také potřeba videozáznam z kamery stabilizovat, aby finální výsledek vypadal lépe. K dispozici je 2D stabilizace, která může kompenzovat skoky a naklonění kamery.

### Základní uzly pro zakomponování scény do reálného záznamu

Pro jednodušší komponování 3D scény do videozáznamu, byly do kompozitoru přidány některé nové uzly. Jsou zde uzly pro 2D stabilizaci, deformaci a reformaci, které je jednoduché použít.

### Neimplementované nástroje

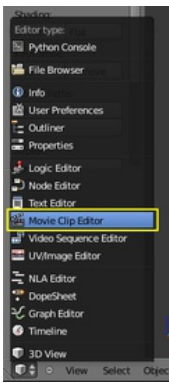
Některé nástroje zatím nejsou v Blenderu dostupné, ale jsou v TODO seznamu. Zatím nejsou podporovány takové věci, jako filtr pohybového zkreslení (rolling shutter filter), řešení pohybu objektu, nebo motion capturing. Nicméně se můžete tyto věci nahradit již implementovanými funkcemi.

## Manuál

### Movie Clip Editor

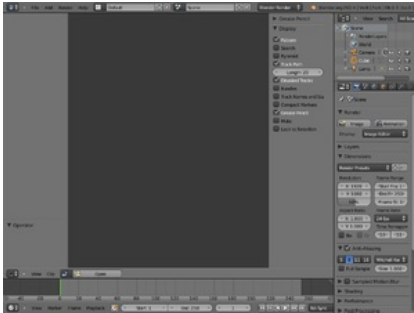
Almost all motion tracking tools are concentrated in editor called Movie Clip Editor. Currently it doesn't have any tools which aren't related on motion tracking, but in future it can be expanded to be used for masking and so, that's why it's called in this more abstract from motion tracking manner.

This editor can be found in list of editor types.



Editor type menu

When you'll switch to Movie Clip Editor interface would change in the following way.



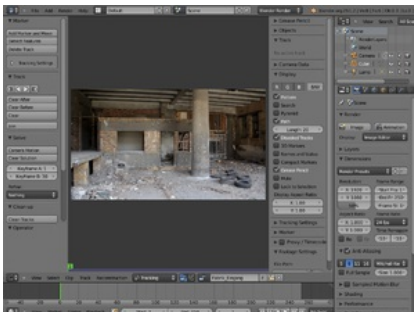
Movie Clip Editor interface

Next thing which is logical to do is to open new video clip to start working with. There are several ways to to this:

- Use **Open** button from movie editor header
- Use Clip » Open menu
- Use AltO> shortcut

Both of movie files and image sequences can be used in the clip editor. If you're using image sequence there's one limitation on naming of files: last group of numbers should be increasing continuously.

So, when movie clip is loaded into clip editor, extra panels are displayed in the interface.



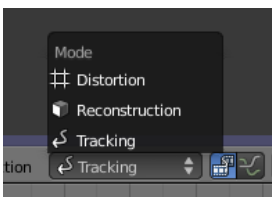
Movie Clip Editor with opened clip

There are plenty of new tools appeared on the screen and here's short description of all of them.

First of all, it should be mentioned, that camera solver consists of three quite separated steps:

- 2D tracking of footage
- Camera intrinsics (focal length, distortion coefficients) specification/estimation/calibration
- Solving camera, scene orientation, scene reconstruction

Tools in clip editor are split depending on step they're used at, so interface isn't messed up with scene orientation tools when only 2D tracking can be done. Currently displayed tools category can be changed using Mode menu which is in editor header.




Movie Clip Editor mode menu


But almost all operators can be called from menus, so it's not necessary to change mode every time you want to use tool which is displayed in panels on differ editor mode.

In tracking mode only tools which are related on tracking and camera solving are displayed. Camera solving tools are exposed here because it's after solving you'll most probably want to re-track existing tracks or place new tracks to make solving more accurate.

### Tools available in tracking mode

#### Markers panel

- **Add Marker and Move** operator places new marker at the position of mouse (which is under the button in this case, not ideal but it's just how things work) and then it can be moved to needed location. When it's moved to needed position, LMB  can be used to finish placing new marker. Also, ↵ Enter and Space can be used to finish placing marker.

But it's faster to use Ctrl LMB  to place markers directly on the footage. This shortcut will place marker on place you've clicked. One more feature here: until you've released mouse button you can adjust marker position moving mouse and using track preview widget to control how accurate marker is placed.

- **Detect Features** operator detects all possible to features on current frame and places markers at this features. This operator doesn't take in account other frames, so it can place markers on features which belongs to moving object and if camera is turning away from this shot, no markers would be paced on frames after camera moved away.

There are several properties for this operator:

**Placement** is used to control where to place marker. By default, they'll be added on the whole frame, but you can also outline some interesting for detecting areas with grease pencil and place markers only inside outlined area. That's how "Inside Grease Pencil" placement variant works. You can also outline absolutely not interesting areas (like trees, humans and so) and place markers outside of this areas. That's how "Outside Grease Pencil" placement variant works.

**Margin** controls distance from image boundary for created markers. If markers would be placed too close to image boundary, they'll fail to track really soon and they should be deleted manually. To reduce amount of manual cleaning up this parameter can be used.

**Trackness** limits minimal trackness for placing markers. This value came from feature detection algorithm and basically it means: low values means most probably this feature would fail to track very soon, high value means it's not much such track. Amount of markers to be added can be controlled with this value.

**Distance** defines minimal distance between placed markers. it's needed to prevent markers placed too close to each other (such placement can confuse camera solver).

- **Delete Track** is quite self-explaining operator which deletes all selected tracks.

#### Track panel

- First row of buttons is used to perform tracking of selected tracks. Tracking can happen (in order of buttons):
  - Backward one frame
  - Backward along the sequence
  - Forward along the whole sequence
  - Forward one frame

This operators depends on settings from Tracking Settings panel which would be described later.

If during sequence tracking algorithm returned failure of tracking some markers, they'll be disabled and tracking will continue for rest of markers. If algorithm returns failure when tracking frame-by-frame, marker wouldn't be disabled and most possible position of feature on new frame would be sued.

- **Clear After** deletes all tracked and keyframed markers after current frame for all selected tracks.
- **Clear Before** deletes all tracked and keyframed markers before current frame for all selected tracks.
- **Clear** clears all markers except current form all selected tracks.
- **Join** operator joins all selected tracks into one. Selected tracks shouldn't have common tracked or keyframed markers at the same frame.

#### Solve panel

**Camera Motion** operator solves motion of camera using all tracks placed on the footage and two keyframes specified on this panel. There are some requirements:

- There should be at least 8 common tracks on the both of keyframes
- It should be well noticeably parallax effect between this two keyframes

If everything came smooth during solve, average reprojection error would be reported to the information space and to clip editor header. Reprojection error means average distance between reconstructed 3D position of tracks projected back to footage and original position of tracks. Basically, reprojection error below 0.3 means accurate reprojection, 0.3-3.0 means quite nice solving which still can be used. Values above 3 means some tracks should be tracked more accurate or that incorrect values for focal length or distortion coefficients were set incorrectly.

**Refine** option specifies which parameters should be refined during solve. Such kind of refining is useful when you aren't sure about some camera intrinsics and solver would try to find best parameter for that intrinsics. But you still have to know approximate initial values - it'll fail to find correct values if they were set completely incorrect initially.

#### Cleanup Panel

This panel contains single operator and it's settings. This operator cleans up bad tracks: tracks which aren't tracked long enough or which failed to reconstruct accurate. Threshold values can be specified from slides below the button. Also, several actions can be performed for bad tracks:

- They can be simply selected

- Bad segments of tracked sequence can be removed
- The whole tracks can be deleted

#### Clip Panel

This panel currently contains the single operator **Set as background** which sets currently editing clip as camera background for all visible 3D viewports. If there's no visible 3D viewports or clip editor is opened in full screen, nothing will happen.

#### Properties available in tracking mode

#### Grease Pencil Panel

It's a standard grease pencil panel where new grease pencil layers and frames can be controlled. There's one difference in behavior of grease pencil from other areas - when new layer is created "by-demand" (when making stroke without adding layer before this) default color for layer would be set to pink. It makes stroke easy to notice on all kinds of movies.

#### Track Panel



Track Panel in clip editor

First of all, track name can be changed in this panel. Track names are used for linking tracking data to other areas like follow track constraint.

Next thing which can be controlled here is marker's enabled flag (using button with eye icon). If marker is disabled, it's position wouldn't be used neither by solver nor by constraints.

Button with lock at the right of button with eye means track is locked. Locked tracks means it can't be edited at all. This helps to prevent accidental changes to tracks which are "finished" (tracked accurate along the whole footage).

Next widget placed on this panel is called "Track Preview" and it displays content of pattern area. This helps to check how good tracking is happening (control there's no slides from original position) and also helps to move track back to needed position. Moving of track can happen directly from this widget by mouse slide.

If anchor is used (position on image which is tracking is differ from position which is used for parenting) preview widget will display area around anchor position. Such configuration helps for masking some things when there's no good feature at position where mask corner should be placed. Details of this technique would be written later.

There's small area below the preview widget which can be used to enlarge vertical size of preview widget (area is highlighted by two horizontal lines).

Next setting is channels control. Tracking happens in grayscale space, so it can be not enough contrast between feature and background to perform accurate tracking. In such cases disabling some color channels can help.

The last thing is custom color and preset for it. This settings overrides default marker color used in clip editor and 3D viewport and it helps to distinguish different type of features (for example features on far plane or near plane and so). Color is also can be used for "grouping" tracks so the whole group of tracks can be selected by color using Select Grouped operator.

#### Camera Data Panel

This panel contains all settings of camera used for screening movie which is currently editing in clip editor.

First of all, predefined settings can be used here. New presets can be added or unused presets can be deleted. But such settings as distortion coefficients and principal point aren't included into presets and should be filled in even if camera presets were used.

- **Focal Length** is already self-described, it's a focal length with which movie was shoot. It can be set in millimeters and pixels. In most cases focal length is given in millimeters, but sometimes (for example in some tutorials in the Internet) it's given in pixels. In such cases it's possible to set it directly in units it's known in.
- **Sensor Width** is a width of CCD sensor in camera. This value can be found in camera specifications.
- **Pixel Aspect Ratio** is a pixel aspect of CCD sensor. This value can also be known from camera specifications, but can also be guessed. For example, you know that footage should be 1920x1080 but images itself are 10280x1080. In this case pixel aspect is:

$$1920 / 1280 = 1.5$$

- **Optical Center** is an optical center of lens used in camera. In most cases it's equals to image center, but it can be differ in some special cases. Check camera/lens specifications in such cases. To set optical center to center of image, there's **Center** button below sliders.
- **Undistortion K1, K2 and K3** is a coefficients used to compensate lens distortion happened on movie shooting. Currently this

values can be tweaked by hand only (there's no calibration tools yet) using tools available in Distortion mode. Basically, just tweak K1 until solving would be most accurate for known focal length (but also take grid and grease pencil into account to prevent "impossible" distortion)

#### Display Panel

This panel contains all settings which controls things displayed in clip editor.

- **Pattern** can be used to disable displaying of rectangles which are correspond to pattern areas of tracks. In some cases it helps to make clip view more clear and check how good tracking is.
- **Search** can be used to disable displaying of rectangles which are correspond to search areas of tracks. In some cases it helps to make clip view more clear and check how good tracking is. Search areas for selected tracks only would be displayed.
- **Pyramid** makes the highest pyramid level be visible. What pyramid itself is would be described later in Tracking Settings panel section, but basically it helps to determine how much track is allowed to move from one frame to another.
- **Track Path** and **Length** controls displaying of path of tracks. So way in which track s are moving can be visible looking at the only one frame. It helps to determine if track jumps from it's position or not.
- **Disabled Tracks** makes possible to hide all tracks which are disabled on current frame. This helps to make view more clear to control if tracking happens accurate enough.
- **Bundles** makes sense after solving movie clip and it works in the following way: solved position of each track gets projected back to movie clip and displayed as small point. Color of point depends on distance of projected coordinate and original coordinate: if they are close enough point would be green, otherwise it'll be red. This helps to find tracks which weren't solved nicely and need to be tweaked.
- **Track Names and Status** displays such information as track name and status of track (if it's keyframed, disabled, tracked or estimated). Names and status for selected tracks is displayed.
- **Compact Markers**. Way in which markers are displayed (black outline and yellow foreground color) makes tracks be visible on all kind of footages (both dark and light). But sometimes it can be annoying and this option will make displaying of marker in more compact manner - outline would be replaced by dashed black lines drawing on top of foreground, so markers areas would be 1px thick only.
- **Grease pencil** means if grease pencil strokes are allowed to be displayed and made.
- **Mute** changes displaying on movie frame itself with black square, It helps to find tracks which are tracked inaccurate or which weren't tracked at all.
- **Grid** (available in distortion mode only) displays grid which is originally orthographic but was affected by distortion model. This grid can be used for manual calibration - distorted lines of grids are equal to straight lines in the footage.
- **Manual Calibration** (available in distortion mode only) applies distortion model for grease pencil strokes. This option also helps to perform manual calibration. More detailed description of this process would be made later.
- **Stable** (available in reconstruction mode only). This options makes displaying frame be affected by 2D stabilization settings. It's only preview option which doesn't actually changes footage itself.
- **Lock to Selection** makes editor be displaying selected tracks on the same screen position along the whole footage during playback or tracking. This option helps to control tracking process and stop it when track is starting sliding off or when it jumped.
- **Display Aspect Ratio** changes aspect ratio for displaying only. It does not affect on tracking or solving process.

#### Tracking Settings Panel

##### Common options

This panel contains all settings for 2D tracking algorithms. Depending on which algorithm is used, different settings would be displayed, but there are also few common for all trackers settings:

**Adjust Frames** controls which patterns are getting tracking, if be more precious, pattern from which frame is getting tracked. Here's an example which should make things more clear.

Tracker algorithm receives two images inside search area and position of point to be tracked in first image. And tracker tries to find position of point from first image on second image.

Now, how tracking of sequence happens. Second image is always image created from frame at which position of marker isn't known (next tracking frame). But different first image can be send to tracker. Most commonly used combinations:

- Image created from frame on which track was keyframed. This configuration prevents sliding from original position (because position which corresponds most to the original pattern is returning by tracker), but it can lead to small jumps and can lead to failures when feature point is getting deformed due to camera motion (perspective transformation, for example). Such configuration is used if **Adjust Frames** is set to 0.
- Image created from current frame is sending as first image to the tracker. In this configuration pattern is tracking between two neighbor frames and it allows to deal with cases of high transformations of feature point but can lead to sliding from original position, so it should be controlled. Such configuration is used if **Adjust Frames** is set to 1.

If **Adjust Frames** is greater than 1, behavior of tracker would be like keyframes for tracks are creating every **Adjust Frames** frames and tracking between keyframed image and next image is used.

**Speed** can be used to control speed of sequence tracking. This option makes nothing with quality of tracking it's just helps to control if tracking happens accurate. In most of cases tracking is happens much faster than realtime playing and it's difficult to notice when track began to slide off from position. In such cases **Speed** can be set to Double or Half to add some delay between tracking two frames, so slide off would be noticed earlier and tracking process can be cancelled to adjust positions of tracks.

**Frames Limit** controls how many frames can be tracked when Track Sequence operator is called. So, each Track Sequence operation would track maximum **Frames Limit** frames. This also helps to notice slide off of tracks and correct them.

**Margin** can be used to make tracks disabled when they are becoming too close to image boundary. This slider controls "too close" in pixels.

### KLT tracker options

KLT tracker is an algorithm used by default. It allows to track most kinds of feature points and it's motion. It's using pyramid tracking which works in the following way. Algorithm tracks larger image than pattern first to find general direction of motion. Then it tracks a bit smaller image to refine position from first step and make final position more accurate. This continues several times. Number of steps of such tracking is equal to **Pyramid Level** option and we tell that on first step tracking happens for highest pyramid level. So Pyramid Level=1 is equal to pattern itself, and each next level doubles tracking image by 2.

Search area should be larger than highest pyramid level and "free space" between search area and highest pyramid level defines how much feature can be moved from one frame to another.

Default settings should work in most of general cases, but sometimes pyramid level should be changes. For example, when footage is blurry, adding extra pyramid levels helps to track them.

This algorithm can fail in situations when feature point is moving in one direction and texture around feature point is moving in another direction.

### SAD tracker options

On each step SAD tracker reviews the whole search area and finds pattern on second image which is mostly like pattern which is getting tracking. This works pretty fast, but can fail in several cases. For example, when there're another feature point which looks like tracking feature point appears in search area. In this case SAD will lead to jump of track from one feature to another.

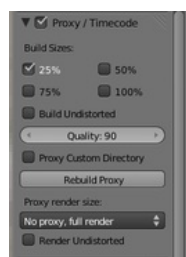
**Correlation** defines threshold value for correlation between two patters which is still considering as successful tracking. 0 means there's no correlation at all, 1 means correlation is full.

There's one limitation: currently, it works for features 16x16 only.

### Marker Panel

This panel contains numerical settings for marker position, pattern and search areas dimensions, offset fo anchor point from pattern center. All sliders are self-explainable.

### Proxy / Timecode Panel



Proxy / Timecode Panel in clip editor

This panel contains options used for image proxies and timacodes for movies.

Proxy allows to display images with lower resolution in clip editor. This can be helpful in cases when tracking of 4K footage is happening on machine with small amount of RAM.

First four options are used to define which resolutions of proxy images should be build. Currently it's possible to build images 25%, 50%, 75% and 100% from original image size. Proxy size of 100% can be used for movies which contains broken frames which can't be decoded.

**Build Undistorted** means that proxy builder would also create images from undistorted original images for sizes set above. This helps to have faster playback of undistorted footage.

Generated proxy images are encoding using JPEG and quality of JPEG codec is controlled by **Quality** slider.

By default, all generated proxy images are storing to <path of original footage>/BL\_proxy/<clip name> folder, but this loction can be set by hand using **Proxy Custom Directory** option.

**Rebuild Proxy** will regenerate proxy images for all sizes set above and regenerate all timecodes which can be used later.

**Use Timecode Index** can (and better be used) for movie files. Basically, timecode makes frame search faster and more accurate. Depending on your camera and codec different timecodes can make better result.

**Proxy Render Size** defines which proxy image resolution is used for display. If **Render Undistorted** is set, then images created from undistorted frame would be used. If there's no generated proxies, render size is set to "No proxy, full render" and render undistorted is enabled, undistortion will happen automatically on frame draw.

### Tools available in reconstruction mode



Proxy / 2D  
Stabilization Panel  
in clip editor

There's one extra panel which is available in reconstruction mode - 2D stabilization panel.

This panel is used to define data used for 2D stabilization of shot. Several options are available in this panel.

First of all, it's list of tracks used to compensate camera jumps, or locaiton. It works in the following way: it gets tracks from list of tracks used for location stabilization and finds median point of all this tracks on first frame. On each other frame algorithm makes this point have the same position in screen coordinates by moving the whole frame. In some cases it's not needed to fully compensate camera jumps and **Location Influence** can be used in such cases.

Camera can also rotate a bit adding some tilt to the footage. There's **Stabilize Rotation** option to compensate this tilt. Extra single track is needed to set for this and it works in the following way. On first frame of movie, this track is getting connected with median point of tracks from list above and angle between horizon and this segment is keeping constant along the whole footage. Amount of rotation applying on the footage can be controlled by **Rotation Influence**.

If camera jumps a lot, it'll be noticeable black areas near image boundaries. To get rid of this black holes, there's **Autoscale** option which finds smallest scale, applying which on the footage all black holes near image boundaries would eliminate. There's option to control maximal scale factor (**Maximal Scale**) and amount of scale applying on the footage (**Scale Influence**)

## Renderovací enginy

- Blender Internal
- [Cycles](#)

## Proces renderování

Renderování je konečná fáze vizualizace v procesu počítačového zpracování grafiky (CG). Je v něm vytvořen 2D obraz vstupní 3D scény. Renderování je vysoce náročný proces na vytížení CPU. Blender je schopen využít i externího renderovacího stroje, nebo farmy strojů, tj skupiny sítí propojených počítačů sdefinovaným komunikačním rozhraním. Tato kapitola vysvětluje funkce Blenderu spojených s procesem výroby obrazů, nebo animací.

Poté, co jste nastavili materiály, textury, osvětlení a kameru, můžete začít s procesem renderování. Je nepravděpodobné, že se první render ihned podaří. Počítejte s tvorbou více testovacích vzorků, při nichž budete ladit poslední parametry pro osvětlení, materiály, časovou osu a podobně.

Blender využívá vlastní renderovací engine. Jde o rychlý renderer s možností produkce kvalitních výsledků. Existuje několik dalších externích renderů, které lze do procesu renderování zapojit jako "externí enginy".

Na celém světě je k dispozici velice různorodá škála výpočetní techniky s převážně odlišnými výkony. Rendering je proces, který dokáže rozkouskovat celkovou práci tak, aby ji zvládl zpracovat i méně výkonný počítač. Na druhou stranu dokáže při náročných parametrech a podmínkách zatížit i více výkonné servery a zaplnit obrovské objemy úložného prostoru. Existuje však cesta kompromisu a tím je renderování pomocí farmy s využitím rozdělení úkolu na malé se zachováním celistvosti.

Tato stránka pojednává o hlavních nastaveních renderovacího stroje na panelu renderu spolu s detailními popisy parametrů a vlastností.

## Přehled

Vykreslování aktuální scény se provádí stisknutím velkého tlačítka image v panelu Render, nebo stisknutím tlačítka F12. Způsob zobrazení výsledku na obrazovce je nastavitelný, viz [Výstup renderu](#), nebo také [Okno renderu](#).

Stiskem tlačítka `{{Literal|Animation}}` se nastartuje produkce celé videosekvence. Výsledkem ztvárnění je ukládán do vyrovnávací paměti a zobrazen v samostatném okně. Může být okamžitě uložen stisknutím tlačítka F3 nebo pomocí menu File->Save Image s nastavením voleb v panelu Output. Animace budou uloženy do výstupního adresáře podle požadovaného formátu. Obvykle jde o sérii snímků, viz část [Možnosti výstupu](#) a [Animace](#). Blender sám podporuje i tvorbu videosouboru s různými video a audiokodeky, tedy je možné render použít jako závěrečný filmový stroj.

Obrazy (video) jsou vykresleny podle rozměrů definovaných v panelu Dimensions - rozměry včetně případného škálování (pro náhled kupříkladu postačuje mimo rozlišení FullHD pouze pětina, tj. 20%. Je vhodné poznamenat, že náhledy renderování jak jsou popsány níže je možné urychlit nejen celkovým rozlišením obrazů, ale vypnutím některých funkcí stínů, materiálových vlastností, raytrace apod. Z pohledu rozlišení při zmenšení výsledného obrazu třikrát bude proces zhruba 9 krát rychlejší).

### Workflow - postup

Obecně proces renderování je následující posloupností aktivit:

1. Vytvoření všech objektů ve scéně.
2. [Osvětlení scény](#).
3. [Umístění kamery](#).
4. Renderování testovacího obrázku o 25%ní velikosti bez převzorkování, nebo ray tracingu atd., což je velice rychlé
5. Nastavení a adjustace materiálů, textur a osvětlení.
6. Iterace těchto kroků dokud nedosáhnete požadovaných výsledků.
7. Renderování s podstatně kvalitnějším plnohodnotným výstupem.
8. Uložení obrázků.

## Distribuovaná renderovací farma

Existuje několik možností alokace výpočetní kapacity CPU které je možné využít pro snížení celkového času renderování.

Za prvé, pokud máte více-jádrový procesor, můžete zvýšit počet podprocesů a Blender bude používat tento počet procesorů pro výpočet renderu (výrazná podpora více vláken a procesů v případě více jader).

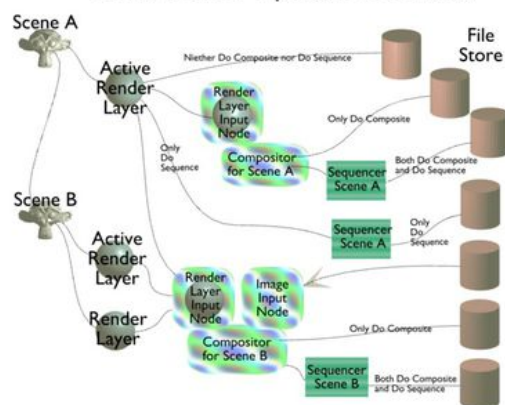
Za druhé, pokud máte lokální síť k dispozici PC, můžete si rozdělit práci až do úrovně jednotlivých počítačů, respektive intervalů videosekvencí. Například pro zpracování 200-snímkové animace na pěti počítačích PC je šikovné rozdělit renderování po úsecích snímků 1-40, 41-80, ... atd. V případě, že je některý počítač méně výkonný je možné mu "přidělit méně práce". Pro tento "kolektivní rendering" je vhodné soubory .blend a zdroje jako textury, mapy umístit na sdílené diskové úložiště viditelné všemi počítači.

Za třetí můžete tak využít WAN renderování, což je způsob, při kterém můžete e-mailem nebo sdílením souborů, nebo Verse - share souboru .blend (ze zabalenými daty!) poslat napříč internetem a využít tak k renderování regionálně odsunutě jiné PC. Lze tak s přáteli kooperovat a vzájemně výpočetním výkonem vypomáhat.

Za čtvrté, můžete použít služeb renderovací farmy. Tyto služby, podobně jako BURP, provozují jednotlivé organizace. Můžete poslat e-mailem soubor, a po té jej distribuovat přes své PC pro renderování. BURP je zde uveden jako nabízená služba zdarma a využívá systém zpracování na pozadí BOINC. Ostatní služby jsou placené, nebo předplatné způsobem "pay-as-you-go" (zaplat jakmile využiješ).

## Integrace pracovního Renderu Workbench

## Some Render Pipeline Possibilities



Blender má tři nezávislé pracovní moduly skrze které jde tok zpracování obrazů v tomto pořadí:

- Rendering Engine
- [Kompozitor](#)
- [Sekvencer](#)

Je možné využít každý výpočetní modul samostatně, anebo v takto propojeném postupu. Například: můžete použít Sekvencer sám o sobě k tomu post-processing na vytvořený video stream. Můžete použít Kompozitor, který sám provede nějaké nastavení barev na obrázku. Můžete vykreslit scénu pomocí aktivní Renderované vrstvy a uložit obrazy přímo - bez použití Kompozitoru nebo Sekvenceru. Tyto možnosti jsou uvedeny v horní části obrázku vpravo.

Můžete také spojit scény a rendery v Blenderu jak je předvedeno, buď přímo, nebo prostřednictvím dočasného ukládání souborů. Každá scéna může mít více vykreslovacích vrstev a každá renderovaná vrstva se skládá v Kompozitoru. Aktivní vykreslení vrstva je vrstva, která se zobrazí a je označena jako aktivní. Pokud není označena jako aktivní, pak je pro výpočet využita v pořadí další renderovaná vrstva. Obraz je zobrazen jako finální render pokud nastavení Compositing a Sequencer NENÍ zapnuto.

Jestliže volba Compositing je povolena, jsou vykreslované vrstvy přiváděny do Kompozitoru (sazeč, skladatel). Z výstupu kompozitoru může být obraz uložen, nebo v případě zapnutí sekvenceru dále zpracováván.

Jestliže volba Sequencer je povolena, je výsledek z skompozitoru (pokud je povoleno "Do Composite") nebo aktivní vrstva renderu (pokud není "Do Composite" povoleno) přiveden na lištu v sekvenceru. Tam je dále zpracováván podle nastavení VSE a nakonec je výsledkem získán obraz dané scény.

Poněkud složitější situace nastává, obsahuje-li asoubor .blend více scén, například scénu A a scénu B. Je-li Compositing zapnuto, uzel renderu ve scéně B může kompozitor vytáhnout jako vrstvu renderu scény A. Tento obraz však není post-procesován. Pokud požadujete vytáhnout výsledek kompozitoru a sekvenceru scény A, je nutné vyrenderovat scénu A do souboru a dále použít uzel vstupu obrazu, který je dále podsouván kompozitoru do scény B.

Spodní část grafických možností ukazuje konečný mix: po zpracování snímků a dynamické komponentě renderu ze scény A je v kompozitoru smíchána se dvěma vrstvami renderu ze scény B, následně sekvencována a uložena jako hotový výsledek.

Tyto příklady jsou jen malou částí všech možností Blenderu. Přečtěte si o všech možnostech a poté užijte svoji kreativitu a vytvářejte své vlastní a jedinečné pracovní postupy

## Panel nastavení renderu

Záložka render obsahuje všechny volby nastavení iterního renderovacího engine, nebo externího, pokud je použitý.

### Render

Zde je možné aktivovat proces renderování [Still Image - statický obrázek](#), nebo [Animation - animace](#).

Také můžete vybrat, který obrázek chcete vyrenderovat, což je popsáno na stránce [Render Display](#).

### Vrstvy

Menu vrstev obsahuje informace pro renderování jednotlivých [vrstev](#) za pomoci [Průchodů renderem](#).

### Rozměry Dimensions

Toto menu obsahuje nastavení pro velikost výsledného obrazu (viz [Vobly výstupu](#)) a volby pro renderování sekvencí obrázků (videoklipů). (viz [Animations](#)).

### Anti-Aliasing

[Antialiasing](#) je důležitý proces pro produkci vysoce kvalitních renderů které nemají zoubkaté kontury, nebo neobsahují jiné schodovité artefakty.

### Motion Blur -

[Motion Blur](#) je důležitý efekt při renderování pohybujících se objektů. Odstraňuje z videa až nerealistické skokové přechody mezi snímky, které zobrazují pohybující se objekty.

## Shading - stínování

Tyto volby jsou pro řízení efektů stínů a kalkulací v renderu. Nevybráním je vypneme.

- [Textury](#)
- [Stíny](#)
- [Subsurface Scattering](#)
- [Mapy prostředí](#)
- [Ray Tracing](#)
- [Řízení barev](#)

Použití lineárního workflow pokud je povoleno.

- [Alfa](#)

Nastavení, jak budou renderovány průhledné pixely.

## Výstup

Nastavení místa pro ukládání výsledku renderu a typ souboru. Viz [Output Options](#).

## Výkon

Control the way the renderer performs with respect to the computer's memory and processor. See [Performance](#).

## Post Processing

Control effects that are applied after the image has been rendered. If you are using the [Compositor](#) or [Sequencer](#), you can tell Blender to process those effects instead of directly rendering the scene.

Ovládání efektů, které jsou použity po okamžiku již vyrenderovaných obrazů. Pokud používáte [Uzly kompozitoru](#) nebo [Sekvencer](#), můžete Blender nastavit pro zpracování těchto efektů namísto vykreslování hotové scény.

Pole jsou využita pokud [renderujeme video](#).

[Dithering](#) je metoda řízeného rozmazávání a prolínání jednotlivých pixelů.

Také je možné povolit [Edge Rendering - rederování hran](#) pro vytvoření efektu kresby, náčrtu, vzhledu kresleného filmu.

## Razítka, označení

[Stamping - razítkování](#) vkládá text přes vyrenderované obrázky velice podobně, jako když se do fotografických souborů PNG, JPEG and EXR ukládají metadata.

## Bake (zapečení)

[Render Baking - zapečení renderu](#) je proces ve kterém se vytváří texturové soubory které obsahují všechny požadované efekty renderu (osvětlení, stíny, barevné informace). Ty jsou velice důležité pro práci s grafikou v reálném čase kdy nemusí nutně docházet ke stále stejným rekalkulacím scény.

## Kamera

Kamera je objekt zajišťující realizaci renderování Blenderem. Definuje která část scény bude viditelná v daný okamžik. Scéna může mít více kamer, ale musí samozřejmě mít alespoň jednu, aby bylo možné vygenerovat obrázek, tj. pohled kamerou.

## Přidání nové kamery

Mód: Object režim

Klávesová zkratka: ⇧ ShiftA přidat nový, F9 změna nastavení.

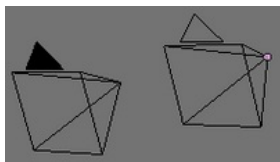
Menu: Add » Camera

V objektovém režimu jednoduše stiskněte ⇧ ShiftA a z vyskakovacího menu vyberte Add » Camera.

## Změna aktivní kamery

Mód: Object mode

Klávesová zkratka: Ctrl0 NumPad



Aktivní kamera (vlevo).

*Aktivní* kamera je kamera, která je právě využívána pro renderovaný pohled, tj. (0 NumPad). Vyberte kameru, která chcete aby byla aktivní a stiskněte Ctrl0 NumPad (také změňte pohled na pohled kamerou).

Aktivní kamera je ve 3D pohledu označena vyplněným trojúhelníkem špicí nahoru viz předchozí obrázek.



Aktivní kamera stejně jako vrstvy může být specifická pro daný pohled, nebo globální (uzamčená) k celkové scéně – viz [část volby pohledu 3D](#).

## Nastavení kamery

Mód: Object mode

Panel: Camera



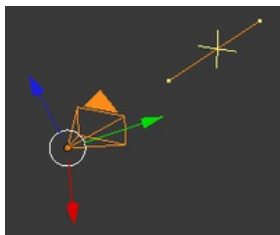
kamery}}.

Kamery jsou pro render neviditelné, a tak nemají žádné nastavení materiálu. Avšak mají panely nastavení stejně jako Object a Editing. Panely jsou zobrazeny v případě, je-li kamera vybrána a je-li aktivní.

## Čočky

- Perspective/Orthographic

Toggles between Orthographic and Perspective modes for the camera. See [the 3D view page](#) for a more detailed description on Orthographic projection, as well as [Doc:2.5/Manual/Render/Perspective|on the next page]].



A camera with the clipping limits and focal point visible.

- Focal Length

Represents the lens focal length, represented in degrees or millimeters. When Orthographic mode is selected, the Focal Length setting changes to the Orthographic Scale setting. This setting determines the size of the camera's visible area.

- Panorama

Renders the scene with a cylindrical camera for panoramic renderings.

- Shift-posuv X/Y

Shifts the camera viewport. Note that most of the time, this setting should not be used to adjust the camera position, as the Shift setting is relative to the actual camera position, which will not be changed.

- Clipping Start/End

Nastavení ohranižení ořiznutím. Jsou renderovány pouze objekty spadající do těchto limitů. Pokud jsou limity povolené, ohraničení klipu je zobrazeno jako dva žluté body na lince pohledu kamery (**C** v obrázku *Camera picture* – první je počátek kamery).

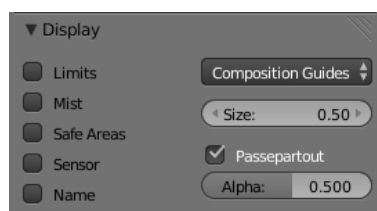
- Hloubka ostrosti

Při použití [hloubky ostrosti](#) bude připojený objekt určovat bod zaostření kamery. Přilinkováno objektu deaktivuje předchozí nastavení ostření kamery.

- Vzdálenost - Distance

Distance to the focal point. It is shown as a yellow cross on the camera line of sight. Limits must be enabled to see the cross. It is used in combination with the [Defocus Compositing Node](#).

## Display



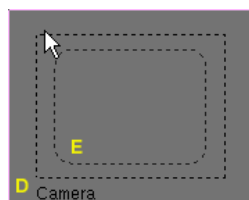
Camera Display panel.

- Limits

Toggles viewing of the limits on and off.

- Mist

Toggles viewing of the mist limits on and off. The limits are shown as two white dots on the camera line of sight (**A** and **B** on the *Camera picture*). The mist limits are set in the World panel, in the Mist section.



Camera View showing Title Safe border and camera name.

- Title Safe

When this is enabled, an extra dotted frame is drawn inside the camera viewport, delimiting the area considered as “safe” for important things, like titles (shown beside **E** in *Camera View*).

- Name-jméno

Toggle name display on and off (**D** on the *Camera Viewpicture*).

- Size-velikost

Size of the camera icon in the 3D view. This setting has no effect on the render output of a camera, and is only a cosmetic setting. The camera icon can also be scaled using the standard Scale S transform key.

- Passepartout, Alpha

Tento mód zajistí stmívání obrazu mimo viditelnou oblast kamery, a to na základě nastavení hladiny Alpha.

### Note

The 3D View window contains settings similar to the camera, such as Orthographic/Perspective and Clip Start/Clip End. These settings have no effect on the camera rendering, and only change the view settings when *not* in Camera view. These settings are accessed through the View menu of the 3D View. See the [3D view options page](#) for more details.

## Composition Guides

Composition Guides are available from the drop-down menu, which can help when framing a shot. There are 8 types of guides available:

- Center
 

Adds lines dividing the frame in half vertically and horizontally.
- Center Diagonal
 

Adds lines connecting opposite corners.
- Thirds
 

Adds lines dividing the frame in thirds vertically and horizontally.
- Golden
 

Divides the width and height into Golden proportions (About 0.618 of the size from all sides of the frame).
- Golden Triangle A
 

Draws a diagonal line from the lower-left to upper-right corners, then adds perpendicular lines that pass through the top left and bottom right corners.
- Golden Triangle B
 

Same as A, but with the opposite corners.
- Harmonious Triangle A
 

Draws a diagonal line from the lower-left to upper-right corners, then lines from the top left and bottom right corners to 0.618 the lengths of the opposite side.
- Harmonious Triangle B
 

Same as A, but with the opposite corners.

## Camera Navigation

Here you will find some handy ways to navigate and position your camera in your scene.

### Note

Remember that the active “camera” might be any kind of object. So these actions can be used e.g. to position and aim a lamp...

### Move active camera to view

Mode: Object mode

Hotkey: CtrlAlt0 NumPad


This feature allows you to position and orient the active camera to match your current viewport.

Select a camera and then move around in the 3D view to a desired position and direction for your camera (so that you're seeing what you want the camera to see). Now press CtrlAlt0 NumPad and your selected camera positions itself to match the view, and switches to camera view.

### Camera View Positioning

By enabling Lock Camera to View in the View menu of the View Properties panel, while in camera view, you can navigate the 3d viewport as usual, while remaining in camera view. Controls are exactly the same as when normally moving in 3d.

### Roll, Pan, Dolly, and Track

To perform these camera moves, the camera must first be *selected*, so that it becomes the active object (while viewing through it, you can RMB  click on the solid rectangular edges to select it). The following actions also assume that you are in camera view (0 NumPad)! Having done so, you can now manipulate the camera using the same commands that are used to manipulate any object:

**Roll:** Press R to enter object rotation mode. The default will be to rotate the camera in its local Z-axis (the axis orthogonal to the camera view), which is the definition of a camera “roll”.

**Vertical Pan or Pitch:** This is just a rotation along the local X-axis. Press R to enter object rotation mode, then X twice (the first press selects the *global* axis – pressing the same letter a second time selects the *local* axis – this works with any axis; see the [axis locking page](#)).

**Horizontal Pan or Yaw:** This corresponds to a rotation around the camera's local Y axis... Yes, that's it, press R, and then Y twice!

**Dolly:** To dolly the camera, press G then MMB  (or Z twice).

**Sideways Tracking:** Press G and move the mouse (you can use X twice or Y to get pure-horizontal or pure-vertical sideways

tracking).

## Aiming the camera in Flymode

When you are in Camera view, the [fly mode](#) actually moves your active camera...

[\[video link\]](#)

## Úvod

V některých situacích je vhodné co nejvíce zvýšit rychlost renderování, nebo přistupovat k Blenderu vzdáleně, nebo vytvořit skript pro renderování více úloh. K tomu je výtečně použitelná možnost parametrického spouštění Blenderu z příkazové řádky.

Jednou z výhod příkazové řádky je, že nevyžaduje prostředí X server (pro neznalé: jde o grafické prostředí v systémech Linux) a s tím spojená výhoda vzdáleného renderování pomocí vzdáleného přístupu k operačnímu systému pomocí protokolu SSH nebo telnet (existují Open Source klienti i pro platformu Windows).

**Poznámka!** Argumenty jsou spouštěny v pořadí, v jakém jsou zadány!

```
blender -b file.blend -a -x 1 -o //render
```

...Nebude pracovat, protože výstup a extenze je nastavena až po té, co byl zadán úkol renderování.

Vždy umístěte parametry **-f** nebo **-a** na řádce jako poslední.

## Syntaxe

```
blender [-b <dir><file>] [-o <dir><file>] [-F <format>]
[-x [0|1]] [-t <threads>] [-S <name>] [-f <frame>]
[-s <frame> -e <frame> -a] [[-P <scriptname>] [-- <parameter>]]
```

### Volby Renderu:

```
-b nebo --background <file>
    Natáhne <file> do pozadí (často používané pro renderování bez uživatelského rozhraní "UI-less")

-a nebo --render-anim
    Renderuje rámce od startu do konce (včetně)

-S nebo --scene <name>
    Nastaví pro renderování aktivní scénu <name>

-f nebo --render-frame <frame>
    Renderuje rámec <frame> a uloží jej.
    +<frame> startuje relativně od pozice frame, -<frame> relativně od konce.

-s nebo --frame-start <frame>
    Nastaví startovní rámec <frame> pro render (použijte před argumentem -a)

-e nebo --frame-end <frame>
    Nastaví koncový rámec <frame> pro render (použijte před argumentem -a)

-j nebo --frame-jump <frames>
    Nastaví počet rámců k přeskočení po každém renderovaném rámu

-o or --render-output <path>
    Nastaví výstupní cestu a soubor.
    Použijte // jako počáteční cestu pro umístění
        relativně k umístění pracovního souboru .blend.
    Znak # je doplněn číslem rámce
        a doplněn nulami.
    ani_##_test.png vytvoří ani_01_test.png
    test-#####.png vytvoří test-000001.png
    Pokud jméno souboru neobsahuje znak #, bude ke jménu
        souboru přidán suffix #####
    Číslo rámce bude přidáno nakonec jména souboru.
    tj.g: blender -b foobar.blend -o //render_ -F PNG -x 1 -a
        //render_ vytvoří //render_####, zapiše rámce jako //render_0001.png//

-E nebo --engine <engine>
    Specifikuje renderovací engine
    pro výpis dostupných enginů spusťte parametr -E help

-t nebo --threads <threads>
    Použije počet <threads> vláken pro renderování a související operace
    [1-64], 0 pro celkový dostupný počet procesorů/jader.
```

### Volby formátu:

```
-F nebo --render-format <format>
    Nastaví formát renderu, platné hodnoty jsou...
    TGA IRIS JPEG MOVIE IRIZ RAWTGA
    AVIRAW AVIJPEG PNG BMP FRAMESERVER
    formáty, které mohou být kompilovány do blenderu, ale
    nejsou dostupné na všech systémech)
    HDR TIFF EXR MULTILAYER MPEG AVICODEC QUICKTIME CINEON DPX DDS

-x nebo --use-extension <bool>
    Nastaví volbu nastavení extenze k souboru nakonci renderu
```

### Volby playbacku:

```
-a <options> <file(s)>
```

```

Playback <file(s)> pracuje pouze, pokud proces neběží na pozadí.
-p <sx> <sy> Otevření s levým spodním rohem na pozici <sx>, <sy>
-m Čte z disku (nebufruje)
-f <fps> <fps-base> Specifikuje FPS (snímky za sekundu)
-j <frame> Nastavení kroku od rámce k rámci
-s <frame> Přehrát od rámce <frame>
-e <frame> Přehrát do rámce <frame>

```

## Volby okna: Options:

```

-w or --window-border
    Otevře okna s rámečkem (implicitní)

-W or --window-borderless
    Otevře okna bez rámečku

-p or --window-geometry <sx> <sy> <w> <h>
    Otevře okno se spodním levým rohem na <sx>, <sy> a šířkou a výškou <w>, <h>

-con or --start-console
    Startuje konzolové okno (ignorováno v případě nastavení parametru -b), (pro Windows)

--no-native-pixels
    Nevyužívá nativní velikost pixelu, pro vysoké rozlišení displejů (MacBook 'Retina')

```

## Specifické volby pro herní engine:

```

-g specifické volby herního engine
-g fixedtime Běží na frekvenci 50 Hertz bez výpadků rámců
-g vertexarrays Používá hranové pole pro renderování (obvykle rychlejší)
-g nomipmap Bez texturového mapování
-g linearmipmap Lineární texturové mapování namísto nejbližšího (implicitně)

```

## Volby prostředí Python:

```

-y or --enable-autoexec
    Povoluje automatické spouštění skriptu python

-Y or --disable-autoexec
    Nepovoluje automatické spouštění skriptu python (scripty pydrivers & startup ), (kompilováno jako nestandardní a implicitní)

-P or --python <filename>
    Spustí skript Pythonu v souboru <filename>

--python-text <name>
    Spustí daný skript Pythonu jako textový blok

--python-console
    Spustí Blender s interaktivní konzolí

--addons
    Čárkou oddělený seznam doplňků (addons) bez mezer

```

## Volby debugování:

```

-d or --debug
    Zapíná režim debugování
    * Tiskne každé volání operátora spolu s argumenty
    * Vypíná zachytávání myši (v některých případech interaguje s debuggerem)
    * Ponechá python sys.stdin spíše nežli jeho nastavení na None

--debug-value <value>
    Nastaví při startu hodnotu <value>

--debug-events
    Zapne debug zprávy pro systém událostí

--debug-handlers
    Zapne debug zprávy pro systém zpracování událostí

--debug-jobs
    Zapne časové profilování pro joby na pozadí.

--debug-python
    Zapne debug zprávy pro python

--debug-wm
    Zapne debug zprávy pro okenní manažer

--debug-all
    Zapne všechny debug zprávy (vyjma libmv)

--debug-fpe
    Zapne vyjímky plovoucí čárky

--disable-crash-handler

```

Vypne crash handler

## Různé volby:

```
--factory-startup
    Přeskočí načítání "startup.blend" v domovském adresáři uživatele

--env-system-datafiles
    Nastaví proměnnou prostředí BLENDER_SYSTEM_DATAFILES

--env-system-scripts
    Nastaví proměnnou prostředí BLENDER_SYSTEM_SCRIPTS

--env-system-python
    Nastaví proměnnou prostředí BLENDER_SYSTEM_PYTHON
-nojoystick
    Vypne podporu joysticku

-noglsl
    Vypne stínování GLSL (shading)

-noaudio
    Přepne zvukový systém na "Žádný"

-setaudio
    Přepne zvukový systém na specifické zařízení
    NULL SDL OPENAL JACK

-h nebo --help
    Vytiskne tuto nápovědu

-v nebo --version
    Vytiskne verzi Blenderu

--
    Ukončuje zpracování voleb. Následující argumenty prochází beze změn.
    Přístup pomocí python knihovny sys.argv
```

## Další volby:

```
/?
    Vytiskne tuto nápovědu (windows)

--verbose <verbose>
    Nastavení úrovně upovídání výpisů.

-R
    Registruje v systému extenzi .blend (pouze Windows)

-r
    Potichu registruje v systému extenzi .blend (Windows only)
```

## Příklady

### Renderování obrázku

```
# blender -b file.blend -o //file -F JPEG -x 1 -f 1
```

- **-b**  
Natáhne Blender bez rozhraní
- **file.blend**  
soubor .blend pro renderování
- **-o //file**  
adresář a cílový soubor obrázku
- **-F JPEG**  
formát obrázku JPEG
- **-x 1**  
doplňí extenzi .jpg ke jménu souboru
- **-f 1**  
renderuje rámeček 1

### Renderování filmu

```
# blender -b file.blend -x 1 -o //file -F MOVIE -s 003 -e 005 -a
```

- **-b**

Natáhne Blender bez rozhraní

- **file.blend**

soubor .blend pro renderování

- **-x**

doplní extenzi .avi ke jménu souboru

- **-o //file**

adresář a cílový soubor filmu

- **-F MOVIE**

uloží film .AVI s nízkou kompresí

- **-s 003 -e 005 -a**

Nastaví start renderování na rámec 003 a konec na 005. **Důležité:** *Můžete použít pouze -s nebo -e, ale musíte dodržet pořadí, jinak Blender nebude pracovat!*

## Spuštění Blenderu se specifickým engine

Parametrem je -E engine nebo --engine engine

```
# blender --engine CYCLES
```

Seznam dostupných renderovacích engine získáte pomocí

```
# blender --engine help
```

nebo

```
# blender -E help
```

Příklad výstupu

```
found bundled python: /home/satishg/bin/blender-2.65a-linux-glibc27-x86_64/2.65/python
Blender Engine Listing:
  BLENDER_RENDER
  BLENDER_GAME
  CYCLES[1]      Done
~/.bin/blender-2.65a-linux-glibc27-x86_64/blender --engine help
```

## Platformy

Jak spustit Blender z příkazové řádky závisí na platformě a na místě, kde je instalovaný Blender. Zde jsou základní instrukce pro různé platformy.

### Windows

Otevřete okno s příkazovým řádkem, jděte do adresáře instalovaného Blenderu a spusťte příkaz

```
# cd c:\<blender installation directory>
# blender
```

### Mac OS X

Otevřete terminálovou aplikaci, jděte do adresáře instalovaného Blenderu a spusťte aplikační balík app příkazem:

```
# cd /Applications/Blender
# ./blender.app/Contents/MacOS/blender
```

Pokud využíváte příkazovou řádku více, je vhodné vytvořit zástupný alias, a tak příkazem 'blender' spustíte terminál. Příklad skriptu je zde:

```
# echo "alias blender=/Applications/Blender/blender.app/Contents/MacOS/blender" >> ~/.profile
```

Pokud nyní otevřete nový terminál, následující příkaz bude fungovat:

```
# blender
```

### Linux

Otevřete terminálovou aplikaci, jděte do adresáře instalovaného Blenderu a spusťte jej příkazem:

```
# cd <blender installation directory>
# ./blender
```

Pokud je Blender nainstalován v proměnné PATH (obvykle je instalován Blender instalačním balíkem), můžete jednoduše využít toto:

# blender

## Renderovací engine Cycles

Renderovací engine Cycles je nový engine dostupný od verze Blenderu 2.61. Prochází stále vývojem a v současné době se zaměřuje na interaktivitu a jednoduchost použití. Samozřejmě bez omezení svých produkčních možností. K dispozici je též [vývojová dokumentace](#).

## Začínáme

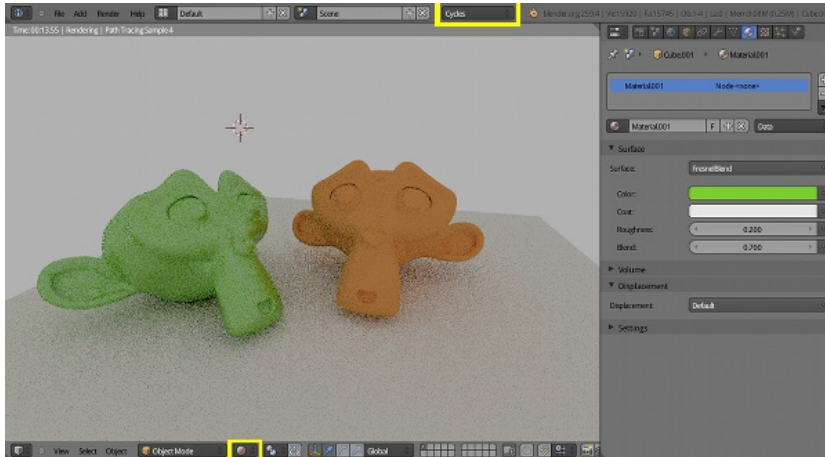
Engine cycles je dodáván spolu v instalačním balíku Blender a implicitně je zapnutý. Pro jeho využití je pouze nutné jej zapnout v horní liště výběru renderovacího engine. Jakmile je zapnutý, je možné ihned interaktivně renderovat nastavením editoru 3D pohledu na kreslicí režim renderování. Render budou udržovat nastavení materiálů a objektů tak, jak je budeme průběžně měnit.

## Odkazy

- [kamera](#)
- [materiály](#)
  - [povrchy](#)
  - [objemy](#)
  - [rozmístění](#)
- [World-svět](#)
- [světla](#)
- [Nodes-uzly](#)
  - [Shaders-stínovače](#)
  - [textury](#)
  - [další...](#)
  - [OSL Shaders](#)
- [renderování vlasů](#)
- [světelné paprsky](#)
- [integrátor](#)
- [redukce šumu](#)
- [průchody renderem](#)
- [editování textur](#)
- [renderování grafickým čipem \(GPU Rendering\)](#)

## Začínáme

Modul renderování pomocí cyklů je dodáván jako doplněk (add on), který je ve výchozím nastavení povolen. Aby bylo možné využít renderování cykly, je nutné mít aktivní nastavení renderovacího motoru (engine) v záhlaví. Interaktivní rendering je možné zahájit pomocí nastavení zobrazení 3D. Render bude doplňovat obraz a aktualizovat jej podle úprav modelářem (materiál, tvar, pohled).



První kroky a nastavení renderování cykly

## Renderování pomocí GPU

Ve speciální dokumentaci dále je popsáno renderování pomocí GPU (procesor na grafické kartě). V případě, že modelář má k dispozici výkonnou grafickou kartu, jejíž mikroprocesor je přímo předurčen pro "renderování za letu", může být renderování mnohem rychlejší nežli při využití výkonu centrálního procesoru. Viz detaily v separátní kapitole [GPU Rendering](#).

## Úvody

je možné shlédnout zde [tutorials](#).

## Tutorials

- [Cycles Complete Overview](#)
- [Cycles Basics Tutorial Series](#)
- [Introduction to Cycles \(Blender Guru\)](#)



- [Introduction to the Cycles Render Engine \(Blender Cookie\)](#)



- [Volumetric Absorption in Cycles](#)



- [Advanced Particle Trail with Cycles](#)



- [An Animated Pin Toy \(BlenderDiplom\)](#)



- [How to create hoar frost in Cycles \(BlenderDiplom\)](#)



- [Create a Realistic Water Simulation \(Blender Guru\)](#)



- [Advanced Organic Material Setup with Maps \(BlenderDiplom\)](#)



- [Rendering a Scene of Wooden Barrels \(Blender Cookie\)](#)



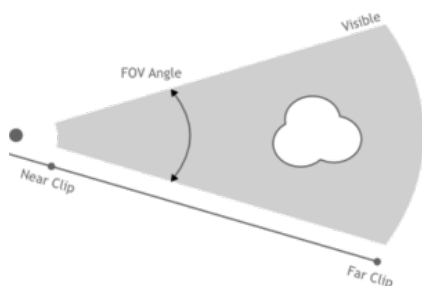
- [Rendering Nuts and Bolts with Cycles \(Free 3D Tutorials\)](#)



Kamera

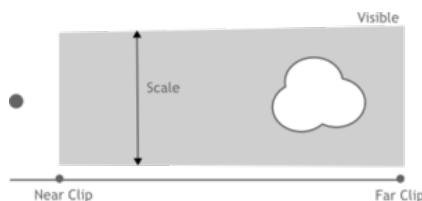
## Perspektiva

Velikost objektivu a úhel (Lens Size and Angle)  
řídí úhel pohledu.



## Ortografické zobrazení

Měřítko (Scale)  
Určuje velikost objektů promítaných do obrázku.



## Panoramatické zobrazení

Cykly podporují panoramatické kamery přímé a typu rybí oko. Všimněte si, že k zobrazení nemůže dojít pomocí renderování metodou OpenGL v pracovním pohledu, ale pouze prostřednictvím vlastního renderování.

### Equirectangular

Render a panoramic view of the scenes from the camera location and use an equirectangular projection, always rendering the full 360° over the X-axis and 180° over the Y-axis.

This projection is compatible with the environment texture as used for world shaders, so it can be used to render an environment map. To match the default mapping, set the camera object rotation to (90, 0, -90) or pointing along the positive X-axis. This corresponds to looking at the center of the image using the default environment texture mapping.

### Rybí oko

Objektivy typu rybí oko jsou obvykle širokoúhlé s výrazným zkreslením. Zobrazení je užitečné pro vytvoření panoramatických snímků, např. projekce typu kulové kamery, nebo také za účelem uměleckého zpracování obrazu. Objektivy typu { Literal | Fisheye Equisolid } bude nejlépe odpovídat skutečné kameře. Poskytuje ohniskovou vzdálenost objektivu a zorné pole úhlu a bude také uvažováno s rozměry vlastního optického senzoru.

Fisheye Equidistant lens does not correspond to any real lens model; it will give a circular fisheye that doesn't take any sensor information into account but rather uses the whole sensor. This is a good lens for full dome projection.

### Lens

Lens focal length in millimeter.

### Field of View

Field of view angle, going to 360 and more to capture the whole environment.

## Hloubka ostrosti (Depth of Field)

### Aperture Type

Method with which to specify the size of the camera opening through which light enters. With Radius the radius of the opening can be specified, while F/Stop specifies the size relative to the camera focal length, a measure more common in photography. Their relation is:  $aperture\ radius = focal\ length / (2\ f-stop)$

### Aperture Size

Also called lens radius. If this is zero, all objects will appear in focus, while larger values will make objects farther than the focal distance appear out of focus.

### Aperture F/Stop

Also called F-number or relative aperture. Lower numbers give more depth of field; higher numbers give a sharper image.

### Aperture Blades

If this setting is 3 or more, a polygonal-shaped aperture will be used instead of a circle, which will affect the shape of out of focus highlights in the rendered image.

### Aperture Rotation

Rotation of the Aperture Blades.

#### Focal Distance

Distance at which objects are in perfect focus. Alternatively, an object can be specified whose distance from the camera will be used.

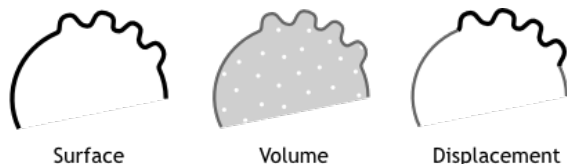
### Clipping

#### Clip Start and End

The interval in which objects are directly visible. Any objects outside this range still influence the image indirectly, as further light bounces are not clipped. For OpenGL rendering, setting clipping distances to limited values is important to ensure sufficient rasterization precision. Ray tracing does not suffer from this issue much, and as such more extreme values can safely be set.

## Materiály

Materiály určují výsledný vzhled sítí, křivek a dalších objektů. Skládají se ze tří shaderů, které definují vzhled povrchu sítě, objemu uvnitř a rozmístění povrchu na síti.



### Shader povrchu

Shader definuje interakci světla na povrchu tělesa. Jedna nebo více BSDF (Distribuční funkce obousměrného rozptylu) určuje, zdali přichodí světlo je odraženo zpět, lomeno směrem k pozorovateli, nebo absorbováno (pohlcováno).

Emise (Emission, vyzařování) určuje jak je světlo vyzařováno povrchem a tedy umožňující, aby každé těleso mohlo být světelným zdrojem.

### Volume Shader

*Shader objemu je v současné době ve vývoji .*

Pokud je shader povrchu neodráží nebo neabsorbuje světlo, prochází světlo objemem. Pokud není žádný objemový shader zadán, bude světlo procházet tělesem až na druhou stranu jeho definiční sítě.

Pokud je definován ,svazek shader popisuje interakce světla při průchodu objemu oka . Světlo může být rozptýlené , vstřebává , nebo emitováno na libovolném místě v objemu .

Materiál může mít jak povrchový, tak objemový shader. Použití jak mohou být užitečné pro materiály jako je sklo, voda nebo led, kde je požadováno, aby část světla byla absorbována v kombinaci se shaderem odlesku na povrchu.

### Zdvihový shader (Displacement)

Depending on the settings, the displacement may be virtual, only modifying the surface normals to give the impression of displacement, which is known as bump mapping, or a combination of real and virtual displacement.

Tvar povrchu a objemu uvnitř může být měněn objemovými shadery. Tímto způsobem mohou být dále textury využity pro vykreslení více detailů.

V závislosti na nastavení může posunutí být virtuální, pouze změnou povrchových normál aby vznikl dojem posuvu. Tento jev se nazývá bump mapping. Také se může využít kombinace reálného a virtuálního posunutí .

## Úspora energie

Systém materiálů je postaven fyzikálních principech s čistým oddělením vzhledu materiálu a který vykreslovací (renderovací) algoritmus je vybrán, což usnadňuje dosažení realistických výsledků a vyváženosti osvětlení. Je třeba však mít na paměti několik skutečností:

In order for materials to work well with global illumination, they should be, speaking in terms of physics, energy conserving. That means they cannot reflect more light than comes in. This property is not strictly enforced, but if colors are in the range 0.0 to 1.0, and BSDFs are only mixed together with the Mix Shader node, this will automatically be true.

K tomu, aby materiály a vystupovat v globální osvětlení, měly by stále platit fyzikální zákonitosti. To znamená, že například tělesa nemohou odrážet více světla nežli na ně dopadá.

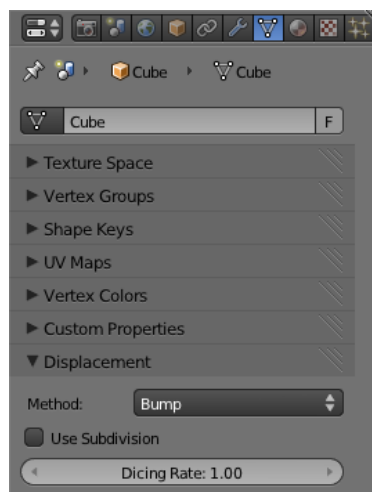
Je však možné toto omezení prolomit například světlem s barevností větší nežli 1,0 anebo pomocí uzlu Přidat Shader. Zde je však namístě opatnost, aby se materiály chovaly přirozeně pro různé směry a další parametry osvětlení.

## Výstupky (Displacement)

Implementace zatím nedokončena. Také značeno jako [experimentální funkce - experimental feature](#).

Tvar povrchu a objemu uvnitř síťoviny může být měněna posunutí shaderů posuvu (displacement). Tímto způsobem mohou být textury využity pro získání větších detailů a podrobností povrchu.

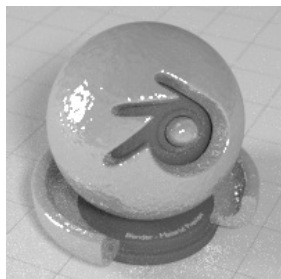
## Typy výstupků



V závislosti na nastavení může posunutí být virtuální, pouze měnící normálu povrchu a tím budící dojem *rozleptání* (známý jako bump mapping), nebo kombinace reálného a virtuálního posunutí. Možnosti typu posunutí jsou :

- *"Pravý zdvih (True Displacement)"* : budou vrcholy posunuty před vykreslováním a upraví tak aktuální síť. Zdvih dává nejlepší kvalitu výsledků v případě jemného dělení sítě (pletiva). Cenou za kvalitu výsledku je však paměťová náročnost operace.
- *'Bump Mapping'* : Při použití povrchového shaderu je využito upravených normál povrchu namísto skutečných normál. Jde o rychlou alternativu skutečného posuvu ale s určitými aproximacemi. Povrchové siluety nebudou zcela přesné a také nevznikne odpovídající stín výtlačků.
- *'Zdvihový + Bump'* : Obě metody lze kombinovat pro posuvy plošek hrubší síťoviny a pomocí bump mapping pro vytvoření konečných detailů.

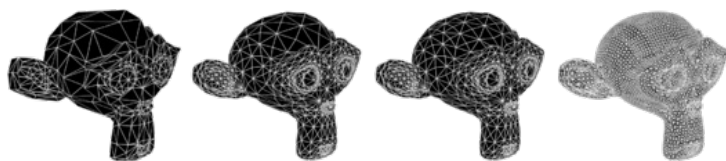
## Podrozdělení - Subdivision



Rozmístění pomocí Bump Map

Implementace není dokončena. Také značeno jako [experimentální funkce](#).

Při použití *Pravého zdvihu* nebo *Zdvihu + Bump* a povoleno *podrozdělení* je možné snížit **hodnotící poměr** rozdělení sítě. Tento projev se týká pouze vykreslení a nezobrazuje se ve výřezu (lze ale ukázat v režimu *Rendered Shading Mode* ). Zdvih je možné také uskutečnit ručně pomocí modifikátoru posuvu.



Podrozdělení vypnuto - zapnuto, poměr dělení 1,0 - 0,3 - 0,05 ( Opička vypadá stejně ve výřezu, bez modifikátorů)

## Povrch (Surface)

Povrchový shader určuje způsob interakci světla dopadajícího na povrch síťoviny objektu. Jedna nebo více funkcí BSDF určuje, zda přichodí světlo je odraženo zpět, láme se k pozorovateli anebo je absorbováno.

Emise definuje, jakým způsobem je světlo vyzařováno povrchem objektu. tím, je možné nastavit, že libovolný povrch může být zdrojem světla.

## Terminologie

- **BSDF** je zkratka pro obousměrnou distribuční funkci rozptylu. Definuje, jak se světlo odráží a láme na povrchu.
- **Reflection (reflexe)** BSDF s odráží přichodí paprsek na stejnou stěnu objektu.
- **Transmission (průchod)** BSDF přenášet přichodí paprsek skrze povrch na jeho opačnou stranu.
- **Refraction (lom)** BSDF je **transmisním** typem a mění směr přichodího světla, které pokračuje na druhou stěnu objektu.

## Parametry BSDF

Hlavním rozdílem od nefyzikálních rendererů je, že odrazy přímého světla od lami a odrazy nepřímého světla dalších povrchů nejsou oddělené, nýbrž se ovládají pomocí jedné funkce BSDF. Což může být lehce omezující, avšak jistě je patrná užitečnost konzistentně vypadajícího povrchu právě jednotným renderem s několika parametry.

For glossy BSDFs, **roughness** parameters control the sharpness of the reflection, from 0.0 (perfectly sharp) to 1.0 (very soft). Compared to **hardness** or **exponent** parameters, it has the advantage of being in the range 0.0..1.0, and as a result gives more linear control and is more easily textureable. The relation is roughly:  $roughness = 1 - 1/hardness$

Pro lesklé BSDFs, '*drsnost* **parametry ovládat ostrost odrazu** , **od 0.0 ( dokonale ostrý ) až 1,0 ( velmi měkká )** . **Ve srovnání s 'tvrdosti 'nebo ' exponent parametry** , to má tu výhodu, že je v rozmezí 0,0 .. 1,0 , a v důsledku toho poskytuje další lineární ovládací prvek , a je snadněji textureable . Vztah je zhruba :  $drsnost = 1 - 1/hardness$

## Objem (Volume)

### Experimentální funkce

Objemové renderování ja zatím rozpracováno. Viz omezení v dolní části této stránky.

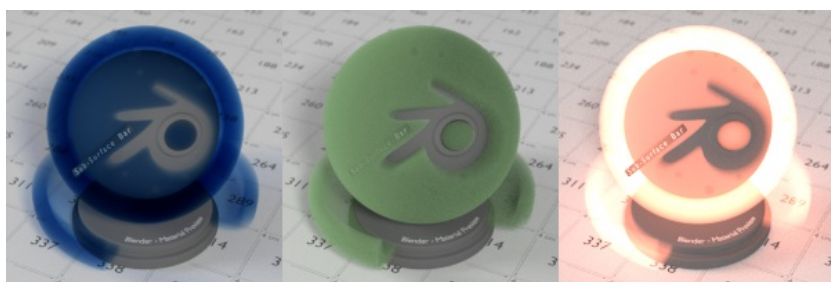
Volume rendering can be used to render effects like fire, smoke, mist, absorption in glass, and many other effects that can't be represented by surface meshes alone.

To set up a volume, you create a mesh that defines the bounds within which the volume exists. In the material you typically remove the surface nodes and instead connect volume nodes to define the shading inside the volume. For effects such as absorption in glass you can use both a surface and volume shader. The world can also use a volume shader to create effects such as mist.

## Volume Shaders

We support three volume shader nodes, that model particular effects as light passes through the volume and interacts with it.

- Volume Absorption will absorb part of the light as it passes through the volume. This can be used to shade for example black smoke or colored glass objects, or mixed with the volume scatter node. This node is somewhat similar to the transparent BSDF node, it blocks part of the light and lets other light pass straight through.
- Volume Scatter lets light scatter in other directions as it hits particles in the volume. The anisotropy defines in which direction the light is more likely to scatter. A value of 0 will let light scatter evenly in all directions (somewhat similar to the diffuse BSDF node), negative values let light scatter mostly backwards, and positive values let light scatter mostly forward. This can be used to shade white smoke or clouds for example.
- Emission will emit light from the volume. This can be used to shade fire for example.



Volume Shader: Absorption / Absorption + Scatter / Emission

## Density

All volume shaders have a density input. The density defines how much of the light will interact with the volume, getting absorbed or scattered, and how much will pass straight through. For effects such as smoke you would specify a density field to indicate where in the volume there is smoke and how much (density bigger than 0), and where there is no smoke (density equals 0).

Volumes in real life consist of particles, a higher density means there are more particles per unit volume. More particles means there is a higher chance for light to collide with a particle and get absorbed or scattered, rather than passing straight through.

## Volume Material

### Interaction with the Surface Shader

A material may have both a surface and a volume shader, or only one of either. Using both may be useful for materials such as glass, water or ice, where you want some of the light to be absorbed as it passes through the surface, combined with e.g. a glass or glossy shader at the surface.

When the surface shader does not reflect or absorb light, it enters into the volume. If no volume shader is specified, it will pass straight through to the other side of the mesh. If it is defined, a volume shader describes the light interaction as it passes through the volume of the mesh. Light may be scattered, absorbed, or emitted at any point in the volume.

### Mesh Topology

Meshes used for volume render should be closed and manifold. That means that there should be no holes in the mesh. Each edge must be connected to exactly 2 faces such that there are no holes or T-shaped faces where 3 or more faces are connected to an edge.

Normals must point outside for correct results. The normals are used to determine if a ray enters or exits a volume, and if they point in a wrong direction, or there is a hole in the mesh, then the renderer is unable to decide what is the inside or outside of the volume.

These rules are the same as for rendering glass refraction correctly.

## Volume World

A volume shader can also be applied to the entire world, filling the entire space.

Currently this is most useful for night time or other dark scenes, as the world surface shader or sun lamps will have no effect if a volume shader is used. This is because the world background is assumed to be infinitely far away, which is accurate enough for the sun for example. However for modelling effects such as fog or atmospheric scattering, it is not a good assumption that the volume fills the entire space, as most of the distance between the sun and the earth is empty space. For such effects it is better to create a volume object surrounding the scene. The size of this object will determine how much light is scattered or absorbed.

## Scattering Bounces

Real world effects such as scattering in clouds or subsurface scattering require many scattering bounces. However unbiased rendering of such effects is slow and noisy. In typical movie production scenes only 0 or 1 bounces might be used to keep render times under control. The effect you get when rendering with 0 volume bounces is what is known as "single scattering", the effect from more bounces is "multiple scattering".

For rendering materials like skin or milk, the subsurface scattering shader is an approximation of such multiple scattering effects that is significantly more efficient but not as accurate.

For materials such as clouds or smoke that do not have a well defined surface, volume rendering is required. These look best with many scattering bounces, but in practice one might have to limit the number of bounces to keep render times acceptable.

## Omezení

V současné době není podporováno:

- Renderování kouře, ohně
- Kamera uvnitř objemu síťoviny
- GPU rendering objemů
- Správná viditelnost paprsků pro objemy

Volume rendering lze použít k tomu, aby účinky, jako je oheň, kouř, mlha, absorpce ve skle, a mnoho dalších vlivů, které nemohou být zastoupeny povrchem oka sám.

Chcete-li nastavit hlasitost, vytvořit síť, která definuje meze, ve kterých se objem existuje. V materiálu, který se obvykle odstraní povrchové uzliny a místo toho připojit hlasitosti uzly definovat stínování uvnitř svazku. Pro účinků, jako je absorpce ve skle můžete využít i plochu a objem shader. Svět může také použít objem shader pro vytvoření efektů, jako je mlha.

## Hlasitosti Shadery

Podporujeme tři objem shader uzly, tento model konkrétní účinky, jako světlo prochází objemu a komunikuje s ním.

- Volume Absorpce absorbuje část světla při průchodu svazku. Toto může být použito k odstínu například černého kouře nebo barevných skleněných předmětů, nebo ve směsi s uzlem objemu rozptylu. Tento uzel je poněkud podobný transparentní BSDF uzlu, blokuje část světla a umožňuje další světlo projít přímo skrz.
- Objem Scatter umožňuje rozptýlení světla v jiných směrech, jak to dopadne částice v objemu. Anizotropie definuje, ve kterém směru je větší pravděpodobnost, že rozptýlí světla. Hodnota 0 nechá rozptýlení světla rovnoměrně ve všech směrech (poněkud podobný difúzní BSDF uzlu), záporné hodnoty at' rozptýlí světla většinou dozadu, a kladné hodnoty nechat rozptýlení světla většinou dopředu. Toto může být použito k odstínu bílého kouře nebo mraků například.
- Emise bude vyzařovat světlo z objemu. Toto může být použito k odstínu ohně například.

[ [ Soubor : cycles\_manual\_materials\_volume.png | frame | center | Objem Shader : Absorpce / Absorpční + Scatter / Emise ] ]

Hustota === ===

Všechny objem shadery mají vstup hustoty. Hustota určuje, kolik světla bude komunikovat s objemem, jak se vstřebává a rozptýlené, a kolik bude procházet přímo skrz. Pro účinků, jako je kouř byste zadat pole hustoty uvést, kde v objemu je kouř a kolik (hustota větší než 0), a tam, kde není žádný dým (hustota se rovná 0).

Objemy v reálném životě se skládají z částic, vyšší hustota znamená, že existuje více částic na jednotku objemu. Více částic znamená, že je větší šance na světlo se srazí s částicí a absorbována nebo rozptýlená, spíše než kolem rovně.

## Objem Materiál

### Interakcí s povrchem Shader

Materiál může mít i povrch a objem shader, nebo pouze jeden ze vstupů. Použití jak mohou být užitečné pro materiály, jako je sklo, voda nebo led, kde chcete, část světla se absorbuje, když prochází přes povrch, v kombinaci s např. sklo nebo lesklý shader na povrchu.

Pokud je povrch shader neodráží nebo absorpci světla, vstoupí do svazku. Pokud není zadán žádný svazek shader, bude to projít přímo skrz na druhé straně sítě. Pokud je definován, svazek shader popisuje interakce světla při průchodu objemu oka. Světlo může být rozptýlené, vstřebává, nebo emitovány na libovolném místě v objemu.

### Mesh topologie

Oka používá pro objem činí by měly být uzavřeny a potrubí. To znamená, že by měl být v síťce žádné díry. Každá hrana musí být připojen k přesně 2 tvářím tak, že zde nejsou žádné otvory nebo T - tvaru plochy, kde se 3 nebo více plochy připojené k okraji.

Normál musí směřovat ven správné výsledky. Normály se používají k určení, zdá se vstoupí nebo opustí objem, a pokud se ukazovat ve špatném směru, nebo tam jedíra v síťovinu, pak renderer je schopen rozhodnout, co je uvnitř nebo vně objemu.

Tato pravidla jsou stejná jako pro správné vykreslování skla refrakce.

## Objem World

Objem shader může být také aplikován na Zcela světa vyplnění celého prostoru .

V současné době je to velmi užitečná pro noční dobu , nebo jiných tmavých scénách , jak svět povrchu shader nebo sluneční lampy nebude mít žádný vliv, pokud je použit objem shader . To je proto, že svět pozadí se předpokládá , že je nekonečně daleko , což je dostatečně přesné pro slunce například . Nicméně pro modelování účinky, jako je mlha nebo atmosférického rozptylu , to není dobrý předpoklad, že objem vyplňuje celý prostor , protože většina vzdálenost mezi Sluncem a Zemí , je prázdný prostor . Pro takové účinky , je lepší vytvořit objekt hlasitosti obklopující scénu . Velikost objektu určí, kolik světla je rozptýleno nebo absorbován .

### Rozptyl skáče

Real World účinky, jako je rozptyl v oblacích nebo podpovrchového rozptylu vyžadovat mnoho rozptylu hopsá . Nicméně objektivní vykreslení těchto efektů je pomalý a hlučný . V typických výrobních film scén může být pouze 0 nebo 1 skáče používá k udržení číni časy pod kontrolou . Efekt vám při vykreslování s 0 Objem odrazí se to, co je známé jako " jediný rozptyl " ,efekt z více odrazů je " více rozptyl " .

Pro vykreslování materiálů, jako je kůže nebo mléka ,podpovrchový rozptyl shader jesblížení těchto různých rozptylových efektů, které je podstatně efektivnější , ale není tak přesná .

U materiálů, jako jsou mraky nebo kouř , které nemají dobře definovanou plochu , je nutné objem vykreslování . Ty vypadají nejlépe s mnoha rozptylu hopsá , ale v praxi , kdo možná bude muset omezit počet odrazů , aby přijatelné činí krát .

## Svítidla

Next to lighting from the background and any object with an emission shader, lamps are another way to add light into the scene. The difference is that they are not directly visible in the rendered image, and can be more easily managed as objects of their own type.

## Svítidla (lamps / lamps)

Spolu s osvětlení vznikajícím na pozadí se světly zářivých těles tvoří svítidla (lamps) dalším způsob osvětlení scény. Lamps nejsou objekty přímo viditelné v renderovaném obrazu, ale je vidět pouze jejich výsledný projev, tj. osvětlení a stíny. Jejich charakteristiky a parametry jsou lehce nastavitelné podobně jako nastavení jiných objektů.

### Druh - Type

V současné době jsou podporovány tyto typy světelných zářičů: **Point** , **Spot** , **Area** a **Sun** . Nejsou podporovány lamps Hemi a budou vykreslovány jako bodové sluneční lamps. V budoucnu je možné tyto typy kvůli kompatibilitě možné využít.

### Velikost - Size

Velikost lamps v jednotkách Blenderu; čím vyšší, tím měkčí stíny a stínování .

### Lité stíny - Cast Shadow

Vypnutím této možnosti nebude světlo vyzařované z lamps blokováno objekty stojící v jeho cestě. Důsledek je urychlení vykreslování (nebudou se sledovat paprsky světelného zdroje).

## Bodové světlo

Bodové žárovky vyzařují světlo rovnoměrně do všech směrů. Při nastavení velikosti - Size větší než nula se stávají kulovitými lampami, které dávají měkčí stíny a stínování. Síla bodových zdrojů je uvedena ve Watech .

## Spot Lamp

Spot lamps vyzařují světlo v určitém směru ohraničeného prostorovým kuželem. Nastavením velikosti - Size větším než nula mohou být měkčí stíny a stínování ještě více plynulejší. Parametr velikost určuje velikost kužele, zatímco parametr směr (blend) může změkčit okraje kužele.

## Plošné světlo (Area Lamp)

Plošná lampy vyzařuje světlo ze čtvercové nebo obdélníkové oblasti pomocí Lambertovské distribuční funkce.

## Sluneční (Sun) světlo

Tyto lamps vyzařují světlo v daném směru. Jejich pozice se nebere v úvahu, protože jsou vždy umístěny mimo scénu v nekonečné vzdálenosti, a tak na osvětlení nemá vliv pozice těles v prostoru. Vzhledem k tomu, že se nenacházejí v scéně, k vyjádření jeho intenzity se využívá různých jednotek, které by měly být obvykle nastaveny na nižší hodnoty než u ostatních světél.

## Uzly (Nódy)

Materiály , světla a pozadí jsou v tomto způsobu renderování definovány pomocí uzlů (nódů). Uzly mají vstupy a výstupy různého typu, od skalárů (čísel), přes vektory (souřadnice), barvy (komponenty) a podobně.

## Shadery

Důležitý koncept pro pochopení tvorby pomocí uzlů a jejich nastavování je **shader socket** ("šuplík stínů"), který popisuje osvětlení jako interakci povrchu nebo objemu tělesa, nikoli jen samotnou barvu tělesa.

Existuje několik typů shaderů, které jsou k dispozici jako uzly:

- **BSDF** shader popisující odraz a lom světla a absorpce světla na povrchu objektu.
- **Emission** shader popisuje emisi - vyzařování světla povrchem nebo objemem objektu.
- **Volume** shader popisuje rozptyl světla uvnitř objemu tělesa.
- **Background** shader popisuje emisi světla z okolního prostředí .

Každý shader uzel má vstup barev a výstup shaderu. Ty mohou být dále míchány sčítány pomocí dalších shader uzlů typu Mix. Zde nejsou uvolněny žádné jiné operace. Výsledný výstup pak může být použit pro renderovací engine jako souhrnný výpočet všech světelných interakcí, pro přímé nebo globální osvětlení.

## Textury

Každý typ textury v režimu Cykly odpovídá jednomu uzlu spolu se souřadnicemi textury a dalšími parametry typu vstup, barvy nebo hodnoty jako výstup. Zde nejsou potřeba žádné textury ve formě datových bloků. Je možné opakovaně znovupoužití nódu pro zpracování textur.

Pro UV mapování a vykreslování textur v pohledu musí být využit obrazový nód. Pokud bude nastaven jako aktivní, vykreslí obraz s režimu textur.

Výchozí souřadnice textur pro všechny nody jsou generované koordináty s výjimkou obrazových textur, které implicitně využívají UV mapování. Každý nód zahrnuje některé volby pro úpravu mapování textur a výsledné barvy. Nastavení je možné vidět ve vlastnostech textury.

## Další

Nody pro geometrická data, souřadnice textur, vrstvení shaderů a "nezyfikální" triky.

## Open Shading Language

Uživatelské nody mohou být vytvořeny pomocí popisného jazyka OSL.

Vstupní uzly

## Kamera

Zobrazit vektor - View Vector

Vektor prostoru viditelnosti kamery od kamery k bodu stínování.

Pohled hloubky Z - View Z Depth

Vzdálenost pohledu - View Distance

Vzdálenost kamery od stínovaného bodu.

## Hodnota

Vstup skalární (jednorozměrné) hodnoty.

Hodnota - Value

Hodnota výstupu.

## RGB

Vstup barvy RGB.

Barva - Color

Barevný výstup RGB.

## Atribut

Získá atribut vztažený k objektu nebo síti. V současné době je možné touto cestou získat UV mapy a barvy vrcholů pomocí jejich jmen. Vrstvy a plánované atributy budou přidány. Touto cestou jsou přístupné vnitřní atributy jako *P* (pozice) , *N* (normály) , *Ng* (geometrické normály) i když chodnější pro přístup jsou uzly.

Name

Jméno atributu.

Color output

Interpolace barvy RGB z atributu.

Vector output

Interpolovaný vektor XYZ z atributu..

Fac output

Skalární hodnota interpolovaná z atributu.

## Geometrie

Geometrické informace o aktuálním stínovaném bodu. Všechny souřadnice vektoru jsou ve světovém prostoru *World Space*. Pro objemové shadery je dostupná pouze pozice a přichází vektor.

Position

Pozice stínící bodu.

Normal

Stínování normál na povrchu (včetně hladkých normál a bump mapping).

Tangent

Tečna na povrchu.

True Normal

geometrická nebo plochá normála povrchu.

Incoming

vektor směřující k bodu stínování je při pohledu na něj.

Parametric

Parametrické souřadnice stínovaného bodu povrchu.

Backfacing

1.0 je-li plocha viděna ze zadní strany, nebo zepředu 0,0.

## Cesta paprsku

Node to find out for which kind of incoming ray the shader is being executed; particularly useful for non-physically based tricks. More information about the meaning of each type is in the [Light Paths](#) documentation.

Is Camera Ray output

1.0 if shading is executed for a camera ray, 0.0 otherwise.

Is Shadow Ray output

1.0 if shading is executed for a shadow ray, 0.0 otherwise.

Is Diffuse Ray output

1.0 if shading is executed for a diffuse ray, 0.0 otherwise.

Is Glossy Ray output

1.0 if shading is executed for a glossy ray, 0.0 otherwise.

Is Singular Ray output

1.0 if shading is executed for a singular ray, 0.0 otherwise.

Is Reflection Ray output

1.0 if shading is executed for a reflection ray, 0.0 otherwise.

Is Transmission Ray output

1.0 if shading is executed for a transmission ray, 0.0 otherwise.

Ray Length output

Distance traveled by the light ray from the last bounce or camera.

Uzel se zjistit , jaký druh přichozího paprsku jeshader vykonán , zvláště užitečné pro triky non - fyzicky založené . Další informace o významu každého typu je v [ [ // Light cesty . | . . . Lehké cesty] ] dokumentace .

Je Fotoaparát Ray výstup

1.0 , pokud stínování se provádí pro ray kamery , 0.0 jinak .

Je Stín Ray výstup

1.0 , pokud stínování se provádí na stínové ray, 0.0 jinak .

Je Difuzní Ray výstup

1.0 , pokud stínování se provádí pro difúzní záření , 0,0 jinak .

Je lesklý Ray výstup

1.0 , pokud stínování se provádí na lesklý ray, 0.0 jinak .

Je Singular Ray výstup

1.0 , pokud stínování se provádí pro singulární ray, 0.0 jinak .

Je Reflection Ray výstup

1.0 , pokud stínování se provádí pro odraz paprsku , 0.0 jinak .

Je Převodovka Ray výstup

1.0 , pokud stínování se provádí pro přenos ray, 0.0 jinak .

Ray Délka výstup

Ujetá vzdálenost od světelného paprsku od posledního odrazu nebo fotoaparátu .

## Object Info

Information about the object instance. This can be useful to give some variation to a single material assigned to multiple instances, either manually controlled through the object index, based on the object location, or randomized for each instance. For example a Noise texture can give random colors or a Color ramp can give a range of colors to be randomly picked from.

Location

Location of the object in world space.

Object Index

Object pass index, same as in the Object Index pass.transformed.

Material Index

Material pass index, same as in the Material Index pass.

Random

Random number unique to a single object instance.

Informace o instanci objektu . To může být užitečné , aby nějaká změna do jednoho materiálu přiřazen více instancí , a to buď ručně ovládané přes index objektu , na základě umístění objektu , nebo náhodně pro každou instanci . Napříkladhuk textura může dát náhodné barvy nebobarvy rampa může dát rozsah barev , které mají být náhodně vybral z .

{ { Literal | Lokalita } }

Umístění objektu ve světovém prostoru .

{ { Literal | Předmět Index } }

Object index průkaz , stejně jako v objektu Index pass.transformed .

{ { Literal | Materiál Index } }

Materiál index průkaz , stejně jako ve Materiál Index průchodu .

{ { Literal | Náhodný } }

Random číslo jedinečné v jedné instanci objektu .

## Fresnel

Dielectric fresnel, computing how much light is refracted through and how much is reflected off a layer. The resulting weight can be used for layering shaders with the Mix Shader node. It is dependent on the angle between the surface normal and the viewing direction.

IOR input

Index of refraction of the material being entered.

Fresnel output

Fresnel weight, indicating the probability with which light will reflect off the layer rather than passing through.

## Layer Weight

Output weights typically used for layering shaders with the Mix Shader node.

Blend input

Blend between the first and second shader.

Fresnel output

Dielectric fresnel weight, useful for example for layering diffuse and glossy shaders to create a plastic material. This is like the Fresnel node, except that the input of this node is in the often more-convenient 0.0 to 1.0 range.

Facing output

Weight that blends from the first to the second shader as the surface goes from facing the viewer to viewing it at a grazing angle.

Dielektrické fresnel , výpočetní , kolik světla láme a kolik se odráží od vrstvy . Výsledná váha může být použit pro vrstvení shadery s Mix Shader uzlu . To je závislé na úhlu mezi povrchem a normálním směru pohledu .

{ { Literal | IOR vstup } }

Index lomu materiálu je zadané .

{ { Literal | Fresnel výstup } }

Fresnel hmotnost , což naznačuje pravděpodobnost , s níž světlo se odráží od vrstvy , spíše než procházet .

## Hmotnost vrstvy (Layer Weight)

Výstupní váhy obvykle používá pro vrstvení shadery s Mix Shader uzlu .

{{ Literal | Blend vstup }}

Blend mezi prvním a druhým shaderu .

{{ Literal | Fresnel výstup }}

Elektrická fresnel hmotnost , což je užitečné například pro vrstvení rozptýlené a lesklé shadery pro vytvoření plastového materiálu . Je to jako Fresnelovy uzlu , kromě toho, že vstup tohoto uzlu je často více , pohodlné 0,0-1,0 rozsahu .

{{ Literal | Tvář v tvář výstup }}

Hmotnost který se hodí zprvního do druhého shaderu jakoplocha se pohybuje od čelem k divákovi zobrazením v úhlu pastvy .

## Souřadnice textur

Commonly used texture coordinates, typically used as inputs for the Vector input for texture nodes.

Generated

Automatically generated texture coordinates from the vertex positions of the mesh without deformation, keeping them sticking to the surface under animation. Range from 0.0 to 1.0 over the bounding box of the undeformed mesh.

Normal

Object space normal, for texturing objects with the texture staying fixed on the object as it transforms.

UV

UV texture coordinates from the active render UV layer.

Object

Position coordinate in object space.

Camera

Position coordinate in camera space.

Window

Location of shading point on the screen, ranging from 0.0 to 1.0 from the left to right side and bottom to top of the render.

Reflection

Vector in the direction of a sharp reflection; typically used for environment maps.

Běžně používané souřadnice textur , které se obvykle používají jako vstupy pro vektorová vstup pro textury uzly .

{{ Literal | Generováno }}

Automaticky generované textury souřadnic z vrcholových pozic síťoviny bez deformace , držet je držet na povrch v rámci animace . Rozsah 0,0-1,0 přes ohraničovací rámeček nedeformované sítě.

{{ Literal | Normal }}

Object prostor normální , pro texturování objektů s texturou zůstat pevně na objekt , protože přetvářejí .

{{ Literal | UV }}

UV textury souřadnic z aktivní omítky UV vrstvou .

{{ Literal | objektu }}

Pozice souřadnic v prostoru objektů .

{{ Literal | Fotoaparát }}

Pozice souřadnic v prostoru kamery .

{{ Literal | Okno }}

Umístění zastínění bodu na obrazovce v rozmezí 0,0 až 1,0 z levé na pravou stranu a spodní do horní části omítky .

{{ Literal | Reflexe }}

vektor ve směru ostrý odraz , typicky používané pro životní prostředí map .

## Informace o částicích

For objects instanced from a particle system, this node give access to the data of the particle that spawned the instance.

Index

Index number of the particle (from 0 to number of particles).

Age

Age of the particle in frames.

Lifetime

Total lifespan of the particle in frames.

Location

Location of the particle.

Size

Size of the particle.

Velocity

Velocity of the particle.

Angular Velocity

Angular velocity of the particle.

Pro objekty důkazem ze systému částic , tento uzel poskytnout přístup k údajům o částice , které plodil instance .

{{ Literal | Index }}

Indexové číslo částice ( od 0 do počtu částic ) .

{{ Literal | Věk }}

Věk částice v rámech .

{{ Literal | Životnost }}

Celková délka života částice v rámech .

{{ Literal | Lokality }}

Umístění částice .

```
{ { Literal | Velikost } }
    Velikost částic .
{ { Literal | Velocity } }
    Rychlost částice .
{ { Literal | Angular Velocity } }
    Úhlová rychlost částice .
```

## Informace o vlasech

This node gives access to strand information.

Is strand

Returns 1 when the shader is acting on a strand, otherwise 0.

Intersect

The point along the strand where the ray hits the strand (1 at the tip and 0 at the root).

Thickness

The thickness of the strand at the point where the ray hits the strand.

Tangent Normal

Tangent normal of the strand.

Tento uzel umožňuje přístup k informacím vlákna .

```
{ { Literal | Je strand } }
    Vráťí hodnotu 1, pokud jeshader jedná o pramen , jinak 0 .
{ { Literal | Intersect } }
    bod podél vlákna , kdepaprsek narazí na pramen (1 na špičce a 0 u kořene ) .
{ { Literal | Tloušťka } }
    tloušťka vlákna v místě, kdepaprsek narazí na pramen .
{ { Literal | Tangent Normal } }
    Tečna normální provazce .
```

## Tangenta (tečna)

Generuje směr tečny pro anizotropních BSDF .

, { { Literal | Směr Type } } :Tangenciální směrový mohou být odvozeny z válcového výstupku kolem X , Y a osy Z (radiální ) , nebo ručně vytvořené UV mapa pro plnou kontrolu .

```
{ { Literal | Tangent Výstup } }
    směr vektor tangenty .
```

Generates a tangent direction for the Anisotropic BSDF.

Direction Type

The tangent direction can be derived from a cylindrical projection around the X, Y or Z axis (Radial), or from a manually created UV Map for full control.

Tangent Output

The tangent direction vector.

## Výstupní nody

Výstupní nody jsou finálními prvky v každém stromu nódů. Je možné jich mít více, avšak pouze jeden aktuální bude použit při renderování (je označen obarvením). Před výstupními nody se vždy nahází stínovače [Shaders](#) vyjma případu naručení povrchu [Displacement](#) nebo jako výstup materiálu.

## [Výstup z materiálu](#)

- Surface: Povrchový výstup z materiálu
- Volume: V současné době nezávislý vývoj. *nedělá nic*
- Displacement: Slouží k vytvoření bump mapping nebo skutečného [podrozdělení](#)

## [Výstup z lampy](#)

- Surface: Není z aktuálního povrchu, ale z konečného výstupu objektu [Lampy](#)

## [Výstup světový \(World\)](#)

- Povrch (Surface): rojev a vzhled prostředí obvykle předchází [stínovač pozadí \(Background Shader\)](#)
- Objem (Volume): *V současné době v rámci nezávislého vývoje , nedělá nic*

Cesty světla - paprsky

## Typy záření

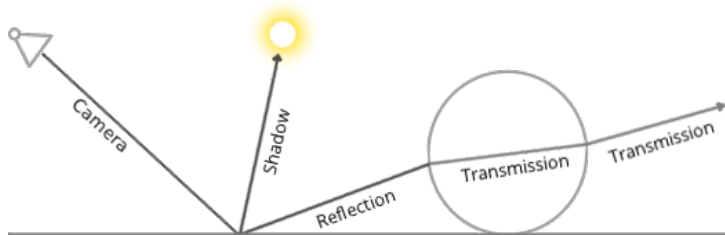
Typy světla (paprsků, záření ) mohou být členěny do těchto čtyř kategorií:

- Kamera: paprsky vycházejí přímo z kamery
- Odraz (reflexe): paprsky jsou vytvářeny odrazem od povrchu
- Transmise: paprsky jsou vytvářeny průchodem skrze povrch
- Stín: záření je využito pro průsvitné stíny

Záření odrazu a průsvitu může dále mít tyto vlastnosti:

- Difuzní (Diffuse): paprsek je generován pomocí difuzního odrazu nebo přenosu (průsvitnost)
- Lesklý (Glossy): paprsek je generován pomocí lesklého zrcadlového odrazu nebo přenos
- Singulární (Singular): paprsek je generován dokonale ostrý odraz a přenos

Uzel cesty paprsku je možné využít k nalezení typu světla pro stínování.



## Ovládání odrazů

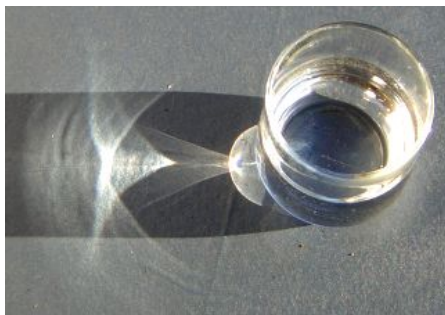
Maximální počet světelných odrazů lze ovládat ručně. Zatímco v ideálním případě by to mělo být počet odrazů nekonečný, v praxi může (musí) stačit menší počet odrazů. Anebo některé lehké interakce mohou být záměrně vynechány pro rychlejší konvergenci výpočtu. Počet difuzních reflexí, lesklých odrazů a odrazů průchodem lze ovládat samostatně.

Cesty světla jsou pravděpodobnostně ukončeny pokud specifikujeme minimální počet světelných odrazů (nižší nežli maximální). V tomto případě bude renderování mnohem rychlejší, avšak výsledný obraz může být zrnitý (zašuměný).

Běžným zdrojem šumu je kaustika, která vzniká rozptylem a následným odrazem světla na hladkých površích. (pokud uvažujeme zdroje paprsků v kameře). Je zde možnost tuto volbu úplně vypnout.

Pojem "kaustika"

Kaustika je optický jev a je tvořená obálkou světelných paprsků odražených, anebo lomených zakřivenou plochou, nebo předmětem. Slovo kaustika také označuje obraz obálky paprsků na jiném povrchu. Běžnou ukázkou kaustiky je sklenice s vodou, na kterou svítí slunce. Sklenice vrhá stín a současně vytváří zakřivenou oblast velmi jasného světla.



Fyzikální popis: pokud souběžné paprsky dopadnou na zakřivenou plochu, jsou dále lomeny v různých směrech. Vlnění z bodového zdroje světla se šíří prostředím s relativní permitivitou  $A$ , a z něho vstupuje do vrstvy, ve které permitivita nejprve klesá do jistého minima a potom opět roste k hodnotě  $A$ . V určité části vrstvy dochází k tomu, že se paprsky obtáčí obloukem zpět ke zdroji. Vrcholy těchto oblouků je možné spojit pomyslnou plochou, přes kterou žádný paprsek neprojde a pod kterou se dva sousedící paprsky protínají. Tato plocha se nazývá kaustika.

## Průsvitnost (Transparency)

Zkratka BSDF

Bidirectional scattering distribution function - distribuční funkce obousměrného rozptylu

Transparentní shader BSDF je předurčen ku zvláštnímu zacházení. Když přes něj prochází paprsek, světlo prochází přímo, jako v situaci, kdy v cestě paprsku nestojí žádný geometrický prvek. Typ světla se průchodem transparentním BSDF nemění.

Alpha průchod výstup je také odlišný pro transparentní BSDF. Další průsvity BSDF jsou považovány za neprůhledné, protože mění směr světla. Stejně tak nemohou být použity pro kompozici alfa, zatímco průsvitnost je možná přes BSDF.

The maximum number of transparent bounces is controlled separately from other bounces. It is also possible to use probabilistic termination of transparent bounces, which might help rendering many layers of transparency.

Maximální počet průhledných odrazů je řízen odděleně od ostatních odrazů . Je také možné použít pravděpodobnostní ukončení průhledných odrazů které by mohly pomoci s renderováním více vrstev průhlednosti.

Všimněte si, že zatímco sémanticky paprsek prochází, jako kdyby žádná geometrie byla zasažena , čímž výkon je ovlivně, protože každý krok transparentnosti vyžaduje provedení shaderu a sledování paprsek .

## Viditelnost

Objekty mohou být nastaveny jako neviditelné pro typ částečného osvětlení:

- Kamera
- Difúzní odraz
- Odlesk (Glossy reflection)
- Průchod (transmisi)
- Stín

Toto může být použito například při vytvoření emitující sítě neviditelnou pro paprsky kamery. Pro případ duplikovaných objektů s dědičností se dědí i vlastnost skrývání objektu před paprsky určitého typu.

Pokud jde o výkon, za použití těchto možností je více efektivní když využívá nastavení uzlu shaderu, který dosahuje stejného účinku. Objekty neviditelné pro určité typy záření budou přeskočen.



### Názvosloví

V českém i v anglickém jazyce je nutné počítat s odlišným významem slov průsvitný (translucent) a průhledný (transparent). Obecně průsvitné prostředí je takové, kterým dokáže projít světlo, zatímco průsvitné prostředí dává světlu možnost průchodu a působí na něj pouze, že jej různými způsoby lomí. Z fyzikálního pohledu jsou si prostředí obou typů velice podobná, avšak z pohledu renderování a především raytracingu je nutné počítat s těmito oběma parametry prostředí. Velice pikantní je z pohledu fyziky optiky i nehomogenní (nesourodé) prostředí, ve kterém se například může světlo šířit jinak podle úhlu průchodu prostředím. Hezkou ukázkou jsou tekuté krystaly a využití v diplejích LCD).

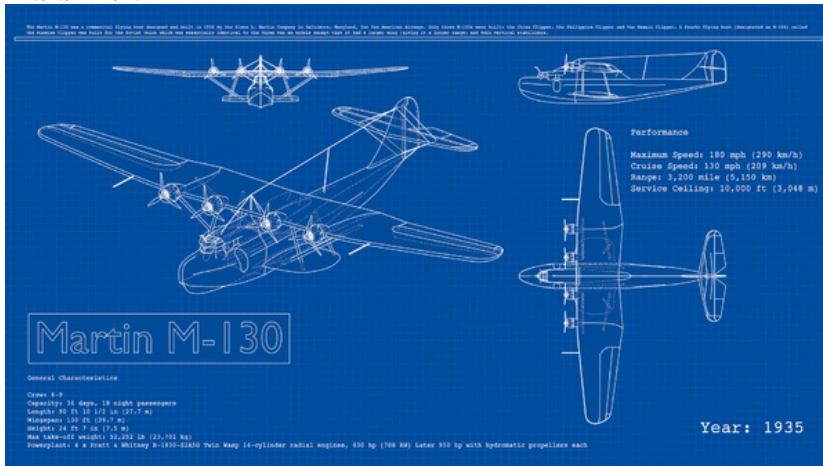
## Co je FreeStyle?

Freestyle je v pojetí grafiky oblast kresby pomocí obrysových hran a čar vykreslované pomocí nefotorealistického renderovacího engine (non-photorealistic NPR). Využívá data tvaru síťového modelu (mesh model) a Z-hloubku pohledu ke kreslení různých typů hran. V nastavení je možné využít různé typy čar a dosáhnout tím zcela ojedinělého a uměleckého ztvárnění "ručních kreseb", "maleb", nebo technických náčrtů.

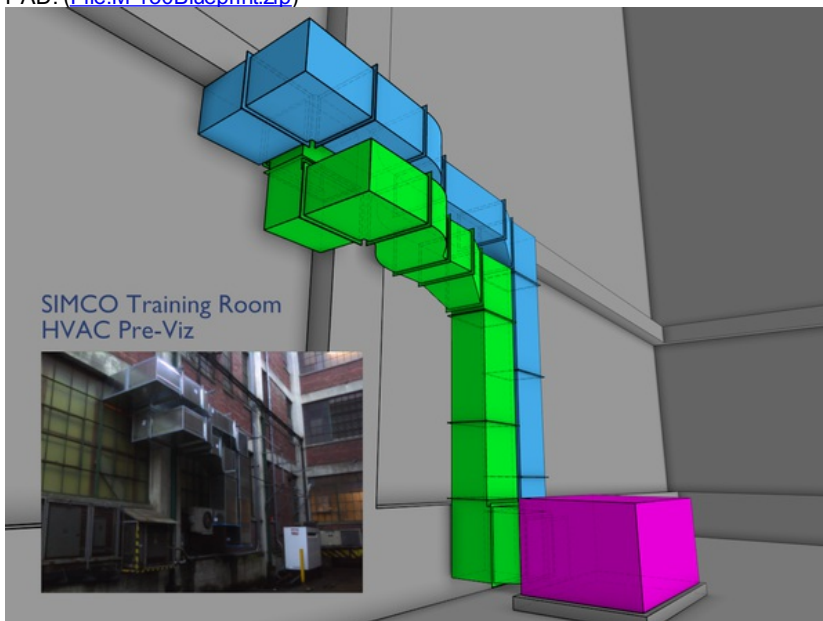
Velkou různorodost stylů čar dávají dva operační módy - [Skriptování Pythonem](#) a [Editor parametrů](#). Styly často mohou pokrývat styly typu japonské kresby velkým štětcem, animované komixy, atp. Těž je možné využít kombinaci různých stylů. Jediným omezujícím faktorem je samo nebe ! pozn. překl.: opravdu v originálu stojí "the sky is the limit!" :-).



Scéna z animovaného filmu [OHA Studio \(the .blend file\)](#). © Mechanimotion Entertainment.



Náčrt letadla Martin M-130 z roku 1935 od LightBWK. CC0. VAROVÁNÍ: OBROVSKÝ SOUBOR! NAVRŽEN PRO STRESS TEST BLENDERU. MŮŽE ZPŮSOBIT JEHO PÁD. ([File:M-130Blueprint.zip](#))



Náčrt vzduchotechniky, Lee Posey. CC0 ([File:HVACPreViz.zip](#))



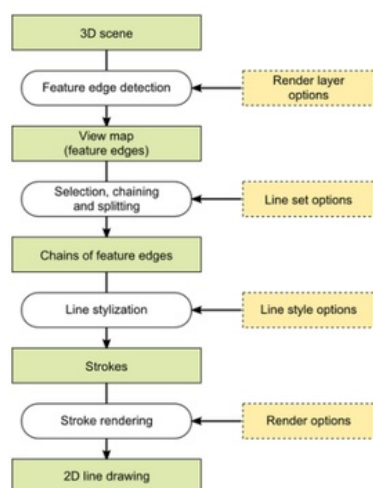
Kuchyně, Vicente Carro. © AnigoAnimation

Více těchto uměleckých lze nalézt na

[http://wiki.blender.org/index.php/Dev:Ref/Release\\_Notes/2.67/FreeStyle#Freestyle\\_Artwork\\_Showcase](http://wiki.blender.org/index.php/Dev:Ref/Release_Notes/2.67/FreeStyle#Freestyle_Artwork_Showcase)

## Základní kontext

1. Aktivujte FreeStyle pomocí okna Buttons → záložka Render → panel FreeStyle a zaškrtněte check box. Renderování FreeStyle je dostupné pouze při využití interního Blender renderu.
2. Nastavení Freestyle je umístěno v novém kontextu Render Layers.
3. Jedna renderovaná vrstva může mít pouze jedinou pohledovou mapu (viewmap). Pohledová mapa sdružuje detekci hran a nastavení (Crease Angle, Culling toggle, Face Smoothness toggle, Material Boundaries toggle, Sphere Radius and Kr Derivative Epsilon advanced options).
4. Pohledová mapa může mít více nastavení čar.
5. Řízení nastavení čar, jejich typů a způsobu renderování ve vaší scéně.
6. Každá čára využívá jeden styl čar (který může být sdílen více sadami čar).
7. Styl čar říká Freestylu jak renderovat skupiny čar v parametrech jako jsou barva, alfa průsvitnost, tloušťka a dalších.

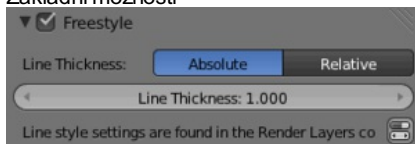


Blokové schéma pohledové mapy Freestyle a procesů

## Známa omezení

- FreeStyle je dostupný pouze pro interní renderer Blender.
- Vysoce náročné na paměť: všechny síťové objekty jsou do paměti nataženy současně.
- Jsou podporovány pouze objekty s plochami. Tyto druhy objektů jsou ignorovány:
  - Síť s drátovými materiály.
  - Síť se zcela průhlednými materiály.
- Průhledné plochy jsou považovány za průsvitné.
- Nejsou detekovány hrany mezi průniky ploch.
- Maskování vrstev nefunguje ve Freestyle.
- Výsledky renderování Freestyle neobsahuje žádnou informaci o hloubce (Z-depth).
- Nepracuje s panoramatickou kamerou.

## Základní možnosti



Základní možnosti renderingu Freestyle.

Aktivujte Freestyle v kontextu Render, kde okno tlačítek Buttons nabízí následující možnosti:

### Line Thickness - Tloušťka čáry

K dispozici jsou dva různé režimy pro definování základní tloušťky čáry:

#### Absolutní

Tloušťka čáry je dána uživatelem zadaný počet pixelů. Výchozí hodnota je **1.0**.

#### Relativní

Jednotka Tloušťka čáry je upravena v poměru k současné vertikální rozlišení obrazu na **480** pixelů. Například pokud jednotka tloušťka čáry je **1.0** s výškou obrazu nastavenou na **480**, potom je **1.5** s výškou **720** a **2.0** s výškou **960**.

### Line Thickness - Tloušťka čáry

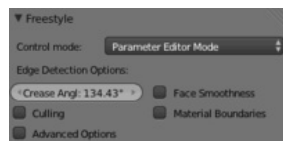
Pouze pro Absolutní tloušťku čáry: základní tloušťka čáry v pixelech je ve výchozím nastavení **1.0**.

## Pohledové mapy - viewmaps

Existuje pouze jedna pohledová mapa na jednu renderovanou vrstvu. Řídí parametry pro detekci hran, které detekované hrany jsou aktuálně renderované a jak. Může být řízena prostřednictvím přívětivého uživatelského rozhraní [editorem parametrů](#), nebo výkonnějším, leč mnohem složitějším [skriptováním pomocí Pythonu](#).

### Hladkost povrchů - Face Smoothness

POkud je povoleno, bude brána v úvahu hladkost v kalkulacích hran.



### Rozhraní editoru parametrů

#### Úhel pomačkání - Crease Angle

Pokud dvě sousední plochy svírají úhel menší, nežli je definováno úhlem Crease Angle, hrany mezi plochami budou renderovány pomocí hrany typu Crease definované v sadě čar. Hodnota má také vliv na hrany siluety (Silhouette).

#### Odpad - Culling

Ignoruje hrany, které jsou mimo pohled (šetří výpočetní čas a paměť, ale může v některých případech snížit kvalitu výsledku).

## Pokročilé možnosti



### Pokročilé možnosti

#### Poloměr koule - Sphere Radius

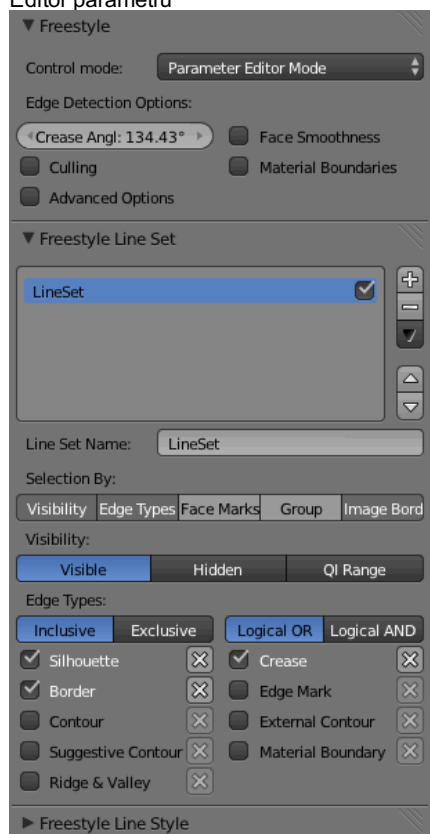
Poloměr má vliv na výpočty křivosti pro hrany na pomezí Ridge, údolí - Valley a kontury - Suggestive Contour a výběru typu čáry z nastavené sady čar.

#### Kr Derivative Epsilon - křivost

Dává kontrolu nad výstupem výběru hran typu Suggestive Contour a Silhouette. Detailnější informace najdete v tomto dokumentu [File:Manual-2.6-Render-Freestyle-PrincetonLinestyle.pdf](#).

Epsilon je údaj z oblasti geometrie a vypovídá o stoupání či klesání křivky. Kr koeficient derivace plochy značí křivost. Vypovídá o "vypouklosti" plochy v daném bodě, o konvexitě, konkávitě.

## Editor parametrů



## Editor parametrů

Editor parametrů metody Freestyle je uživatelsky přívětivé rozhraní pro definici a řízení výběru sady čar a stylů při renderování.

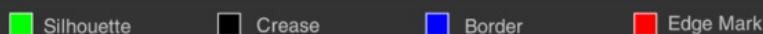
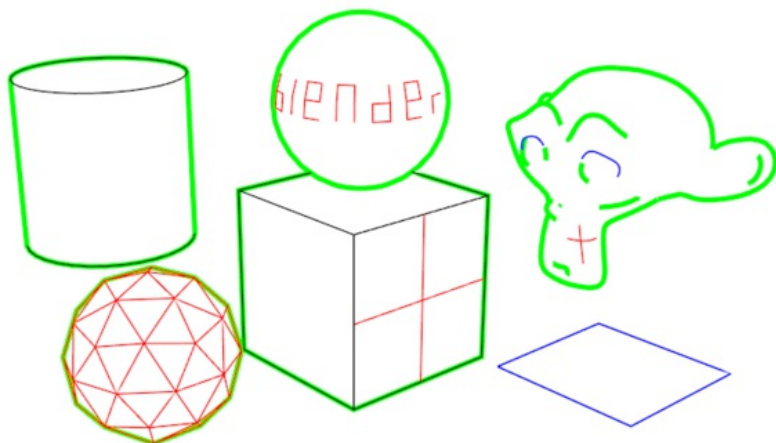
[Sady čar](#) řídí, které z detekovaných hran budou renderovány.

[Styly čar](#) řídí, jak budou vybrané hrany renderovány.

Pohledová mapa (tedy render vrstva) může mít více řádek nastaví, kde každá sada čar a linií je provázána k stylům kreslení čar.

## Sady čar - line set

Sady čar vybírají mezi linkami (hranami) detekovanými pomocí Freestyle, která z nich bude renderována jakým [stylem čáry](#), a to různými metodami.



Příklady některých základních typů hran - od LightBWK ([File:EdgeType.zip](#))

## Výběr podle Viditelnosti

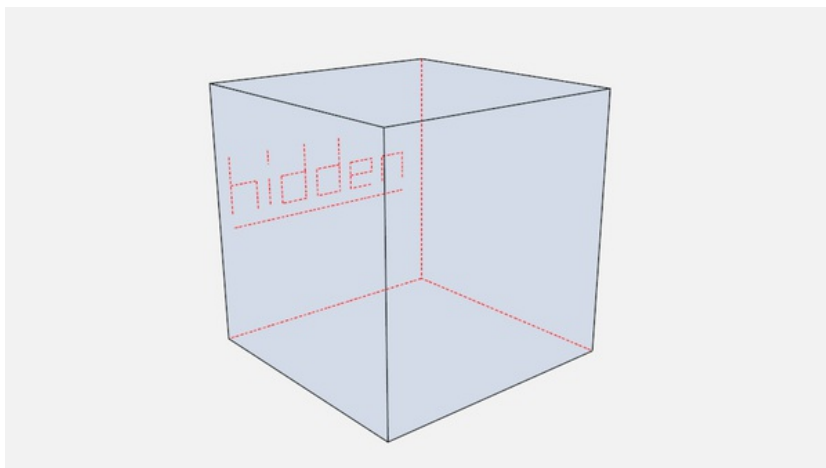
K dispozici jsou tři možnosti pro výběr hran podle viditelnosti.

### Visible-viditelná

Jsou vykresleny pouze okluzní linie.

### Hidden-skrytá

Jsou vykresleny linky ines s alespo'N jedním podvrchem.



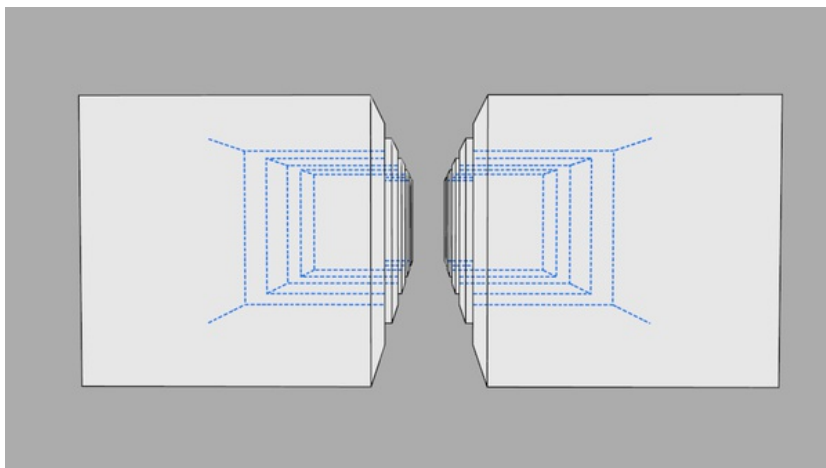
Ověření viditelných a skrytých hran - od LightBWK ([File:HiddenCreaseEdgeMark.zip](#))

### QI Range-QI rozsah

QI je zkratkou pro *Kvantitativní neviditelnost*. Jsou renderovány okluzní linky s počtem ploch v daném rozsahu.

### Start a End-konec

Renderovány linky pouze s QI Range, min/max počet povrchů..



QI Range demonstrace funkčnosti, Start: 3, End: 7, - od LightBWK ([File:QI-Range.zip](#))

## Výběr podle typů hran

Edge types are basic algorithms for the selection of lines from geometry. When using the parameter editor you have to choose at least one edge type in order to get a render output, but several edge types can be combined in one line set. Edge types can also be excluded from calculation by pressing the X next to them.

## Výběr podle typy hran

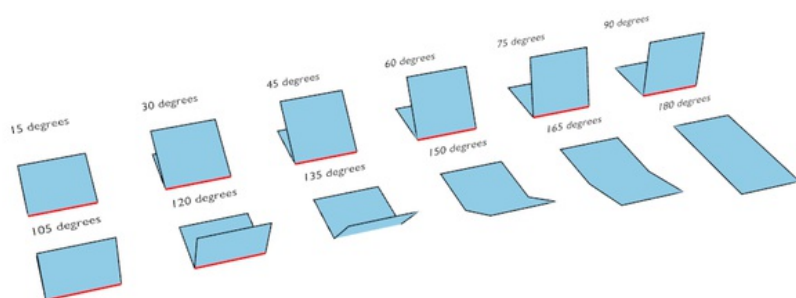
Typy hran patří k základním vstupům algoritmů pro výběr čar. Při použití parametru editoru si musíte vybrat alespoň jeden typ hrany, aby bylo možné získat na výstupu renderování. Pro jednu výslednou čáru je možné kombinovat několik typů hran. Z výpočtu je možné také určité typy hran vyloučit a to pomocí stisku tlačítka `{{Literal|X}}` vedle každé z nich.

### Silhouette

Draws silhouettes around your closed objects; it is often good for organic objects (like Suzanne & Sphere), and bad for sharp edges, like a box. It can't render open mesh objects like open cylinders and flat planes. The output is affected by the Kr Derivative Epsilon viewmap setting.

### Crease

Shows only edges whose adjacent faces form an angle greater than the defined viewmap's Crease Angle.



Crease Angle proof of concept for 121° by LightBWK ([File:CreaseAngle.zip](#))

### Border

Border is for open/unclosed edge meshes; an open cylinder has an open edge at the top and bottom, and a plane is open all around. Suzanne's eye socket is an open edge. All open edges will have lines rendered. This depends on the mesh structure.

### Edge Marks

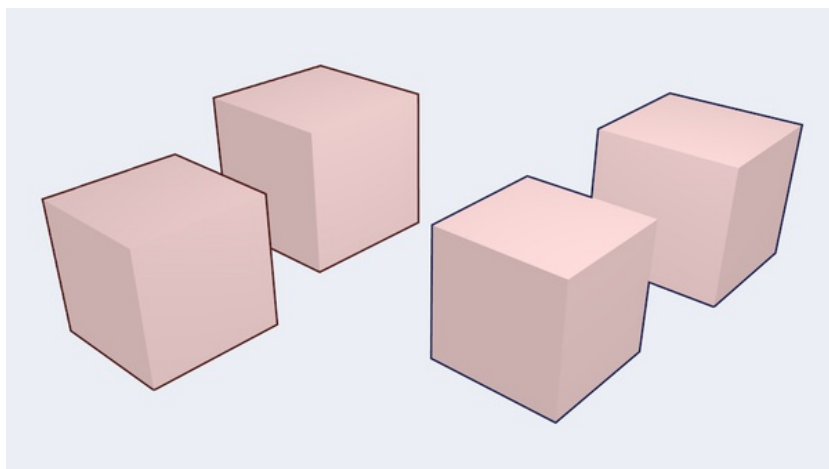
Renders marked edges. See [below](#) for details.

### Contour

Draws the outer edges and inner open border.

### External Contour

Draws the contour lines, but only on the outer edges.



Left pair: Contour; Right pair: External Contour

### Suggestive Contour

Draws some lines which would form the contour of the mesh if the viewport was shifted. Depends on your viewmap settings for Kr Derivative Epsilon and Sphere Radius (further information: [File:Manual-2.6-Render-Freestyle-PrincetonLinestyle.pdf](#)).

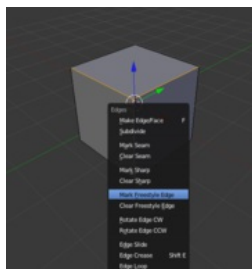
### Material Boundary

Draws lines where two materials meet on the same object. Must be activated in the viewmap settings.

## Ridge & Valley

Draws ridges and valleys. Depends on your Sphere Radius viewmap settings.

## Edge Marks



Select and mark Freestyle edges.



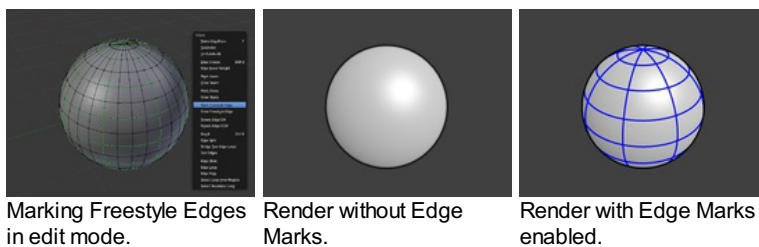
Edge Mark setting in the Line Sets tab.

In edit mode you can mark "Freestyle Edges" in the same manner you can mark "Seams" for UV unwrapping or "Sharp" for edge split. These marked edges are available to render when you select Edge Mark.

This is done as follows:

- Select your mesh and tab into Edit mode.
- Select the edges you want to be marked.
- Press **Ctrl+E** and select Mark Freestyle Edge.

Edge marks are useful when you want to draw lines along particular mesh edges. The examples below explain the use of edge marks.



Marking Freestyle Edges in edit mode.

Render without Edge Marks.

Render with Edge Marks enabled.

The image on the left shows a sphere in Edit mode. The green lines are the edge marks. On the right you see a render without edge marks enabled.

With edge marks enabled, the previously-marked lines are always rendered. You can see the black contour lines and the blue lines that are made with edge marks.

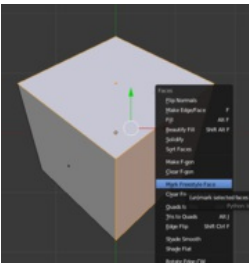
What are edge marks good for?

1. When you need to render marks on an almost-flat plane, when other edge types can't detect any line.
2. When you want full control of edge rendering. Often used for edges of squarish shapes.
3. Mark the whole base mesh to be rendered for base mesh preview.

What are edge marks not good for?

1. Round outer edges (use instead Contour/External Contour/Silhouette).

## Selection by Face Marks



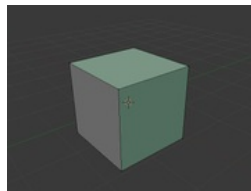
Mark Freestyle Faces.

To set a face mark:

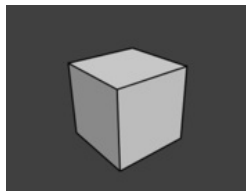
- Select a mesh and tab into Edit mode.
- Select the faces you want to be marked.
- Press **CtrlF** and select **Mark Freestyle Face**.

Face marks are useful for removing lines from certain areas of a mesh.

In this example, two faces of the default cube are marked like the image on the left. On the right is a render without face marks activated.



Marked Faces.



Render Output.



Face mark options.

The line selection can be controlled via inclusion and faces options:

#### Inclusive/Exclusive

Whether to include or exclude edges matching defined face mark conditions from the line set.

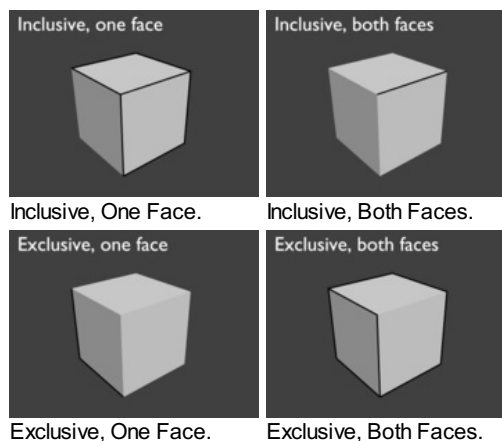
#### One Face

(De)select all edges which have one or both neighbor faces marked.

#### Both Faces

(De)select all edges which have both of their neighbor faces marked.

The image below shows the resulting combinations.



Inclusive, One Face.

Inclusive, Both Faces.

Exclusive, One Face.

Exclusive, Both Faces.

## Selection by Group

You can include or exclude objects for line calculation, based on their belonging to a group.

#### Group

The name of the object group to use.

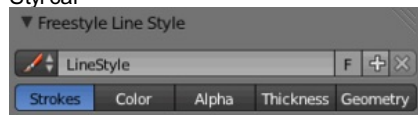
Inclusive/Exclusive

Whether to include or exclude lines from those objects in this line set.

## Selection by Image Border

If enabled, Freestyle only takes geometry within the image border into consideration for line calculation. This reduces render times but increases continuity problems when geometry is moved out of and into camera view.

## Styl čar



### Styl čáry - panel nastavení

Ve Freestyle nastavení stylu čáry a vůbec celkovou grafickou charakteristiku výsledné čáry stanovuje pět nezávislých aspektů:

[tahy](#)  
[barva](#)  
[alfa](#)  
[tloušťka](#)  
[geometrie](#)

Ve vzájemné kombinaci s různými nastaveními umožňují získat veliké množství různých kresbových stylů. Technické kreslení, hrubý náčrtek, karikatura, orientální kaligrafie, atd. patří mezi známé.

Můžete si tak vytvořit mnoho stylů čar a dané styly znovu používat výběrem pomocí rozbalovací nabídky vedle názvu stylu.

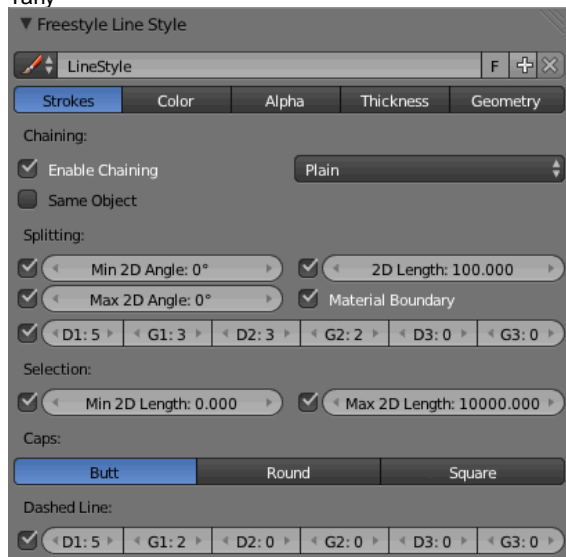
### Délková jednotka

Pokud není uvedeno jinak , všechny délky v nastavení stylu jsou uváděny v pixelech (relativní nebo absolutní, jak je stanoveno v [core options - hlavní volby](#)).



demonstrace různých stylů čar. [File:LineStyle.zip](#)

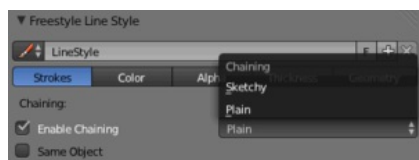
## Tahy



Styl tahových kreslicích linií

Tahy jsou konečnými vykresleními čar. Je dále možné je vylepšit mnohými způsoby. Například odstraněním delších, či kratších tahů ve srovnání s určitou prahovou hodnotou, navázané tahy spojit do jednoho nebo rozbit celistvý tahu do několika kratších podle svírajícího úhlu. Využití z nich na základě úhlu, čárkovaný vzor, atd.

## Řetězení - Chaining



řetězení - Chaining

Ve výchozím nastavení jsou všechny linky ze sady linek zřetězeny dohromady. Existují dvě základní metody řetězení:

obyčejné -Plain

Výchozí metoda zřetězení vytvářející jednoduché řetězce.

Sketchy

Tato volba umožňuje řetězení generování řetězců k vykreslení hran několika tahy. V podstatě se vytváří zakulacené - Round črty místo jednoduché linky. Toto je užitečné pokud používáte některé náhodné modifikátory pro styl čar.

Rounds

Určuje počet koleček povrchních tahů.

Řetězení je možné také vypnout, a tím nechat vykreslovat každou linku zvlášť, což může být vhodné pro styly čar záviselých na přesnosti reprezentace skupiny čar.

## Půlení - Splitting



Půlení - Splitting

Je možné rozdělit řetězce linek Freestyle pomocí těchto kontrolních údajů:

Materiálové hranice - Material Boundary

rozdělí řetězce v případě přechodu jednotlivých materiálů.

Minimální úhel 2D Angle a maximální úhel 2D

Rozdělí řetězce jež svírají rovinný úhel v daném intervalu úhlů.

Délka 2D

Rozdělí řetězce pokud jsou delší nežli daná hodnota.

D1/G1/D2/G2/D3/G3

Rozdělí řetězce pomocí přerušovaného vzoru, v němž "D" je zkratka pro pomlčku a "G" pro mezeru. Rozdělí řetězec za použití tečkovaného vzoru, viz také [níže](#).

## Výběr



## Výběr

Je možné si vybrat řetězce delší než Min 2D Length, nebo kratší než Max 2D Length.

## Zakončení



### Ukončení čar

Můžete si vybrat mezi třemi typy zakončení linek:

#### Butt

Flat cap, exactly at the point the line ends.

#### Round

A half circle centered on the end point of the line.

#### Square

A square centered on the end point of the line (hence, like the circle, the drawn end of the line is slightly extended compared to its computed value).

## Čerchovaná čára - Dashed Line



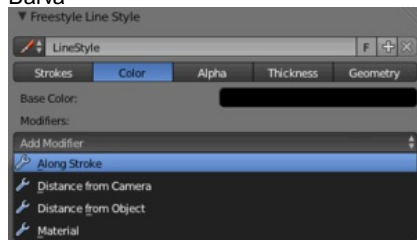
### Rozhraní nastavení čerchování

Pro povolení čerchovaných linek zaškrtněte políčko. Pak můžete určit tři páry teček-čárek a mezer. Hodnoty "dash" definují délku čárky, zatímco hodnoty mezer určují intervaly mezi dvěma pomlčkami.

Pokud je nastavena mezera nulová, pak odpovídající pomlčky jsou ignorovány, přestože mají nenulovou hodnotu.

Pomlčky jsou považovány za samostatné tahy, což znamená, že je možné využít zakončená čar, stejně jako barvu, alfu a modifikátory tloušťky.

## Barva



Karta nastavení barvy tahů

Na této kartě je možné ovládat barvu vašich tahů.

Základní barva

Základní barva pro tento typ linky.

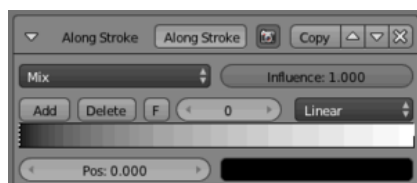
## Modifikátory

K dispozici jsou čtyři modifikátory barvy, které mohou být promíchány s podkladovou barvou pomocí obvyklých metod (viz např. [Mix kompozitních uzlů](#)). Stejně jako u jiných modifikátorů platí pořadí jejich vyhodnocování podle zásobníku shora dolů.

Vliv - Influence

Jak velký vliv má modifikátor na aktuální barvu.

## Podélné tahy

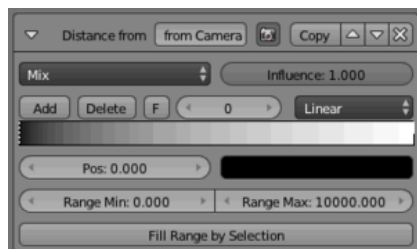


Modifikátor barvy podélných tahů

The Along Stroke modifikátor mění základ barvy s novou, která je daná barevným přechodem podél délky tahu. Jinými slovy aplikuje změnu barvy podél čáry.

Zde je nastavení Blenderovského barevného přechodu.

## Vzdálenost od kamery



Modifikátor změny barvy podle vzdálenosti od kamery

Vzdálenost od kamery color modifier alters the base color with a new one from a given color ramp, using the distance to the active camera as the parameter.

je hlavním parametrem pro změnu barvy čáry pomocí přimíchané barvy z nastaveného spektra barev.

Rozsahy Range Min a Range Max

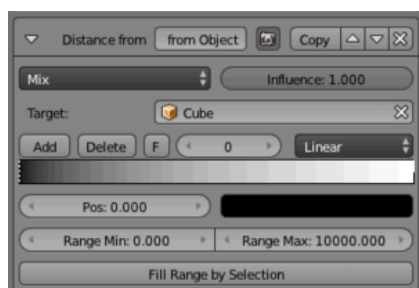
Rozsahy vystupují jako meze mapování parametru "vzdálenost kamery" do výsledného "barvy ze spektra". Pokud vyhovuje je aktuální kreslicí bod vzdálen Range Min, nebo méně, je vybrána jako kreslicí barva počáteční v zadaném barevném rozsahu. A naopak, je-li ve vzdálenosti Range Max a větší, bude mít barvu koncové části spektra. Tyto hodnoty jsou v délkových jednotkách dané scény, nikoli v pixelech!

Vyplnění rozsahu výběrem - Fill range...

Nastaví hodnoty rozsahu min/max podle vzdálenosti vybraných objektů ve scéně.

Ostatní nastavení jsou standardního typu "barevný rozsah" barev Blenderu.

## Vzdálenost od objektu



Modifikátor stylu linky podle vzdálenosti od objektu

Modifikátor Vzdálenost od objektu umožňuje řídit barvu vykreslované linky závislosti na vzdálenosti od daného objektu.

**Cíl - Target**

Objekt od něhož je měřena vzdálenost.

**Rozsahy Range Min a Range Max**

Rozsahy vystupují jako meze mapování parametru "vzdálenost od objektu" do výsledného "barvy ze spektra". Pokud vyhovuje je aktuální kreslicí bod vzdálen Range Min, nebo méně, je vybrána jako kreslicí barva počáteční v zadaném barevném rozsahu. A naopak, je-li ve vzdálenosti Range Max a větší, bude mít barvu koncové části spektra. Tyto hodnoty jsou v délkových jednotkách dané scény, nikoli v pixelech!

**Vyplnění rozsahů výběrem - Fill range...**

Nastaví hodnoty rozsahu min/max podle vzdálenosti vybraných objektů ve scéně a cíle.

Ostatní nastavení jsou standardního typu "barevný rozsah" barev Blenderu.

## Materiál



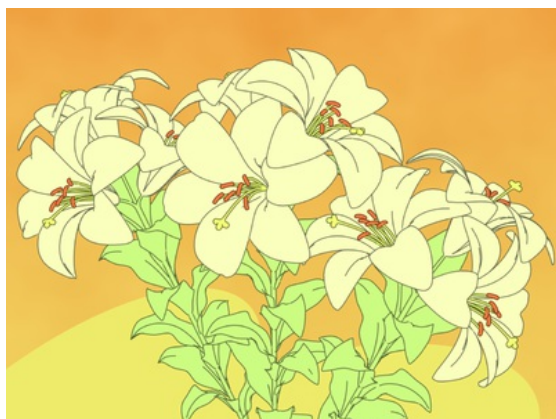
Modifikátor barvy stylu čáry

Modifikátor barvy materiálu mění základní barvu a přimíchává novou, která je brána z aktuální barvy materiálu pod místem vykreslování čáry.

Můžete využít různých vlastností materiálů, z nichž mnohé jsou monokomponentní (tj. dávají černobílé výsledky). V tomto případě může být mapování nastaveno na změnu škály šedi do škály barevnosti.

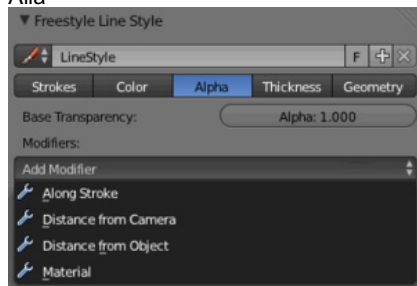
Je-li použita s {{ Literal | Splitu Materiál }} možnosti {{ vDoslovný | zdvihu }} karta, bude výsledek nebude rozmazaný mezi materiály podél tahy.

Je-li použita volba Split by Material- rozdělit podle materiálu v záložce Stroke, výsledek nebude mezi materiály podél kreslených tahů rozmazaný.



Demo materiálového modifikátoru podle T.K. [File:Lilies Color Material.zip](#)

## Alfa



Záložka stylu linky alfa

Na této kartě můžete kontrolovat alfa (průhlednost) obrysových tahů.

Základní průhlednost - Base Transparency

Hodnota základní alfa pro tento daný styl čáry.

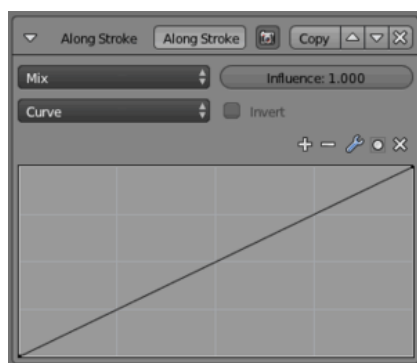
## Modifikátory

K dispozici jsou čtyři alfa modifikátory, které mohou být míchány se základním nastavením alfa pomocí skupiny obvyklých metod (pro detailnější popis viz např. [Mix compositing node - kompoziční uzel](#)). Stejně jako dalších modifikátory jsou aplikovány shora dolů.

Influence - vliv

Úroveň ovlivnění výsledné průhlednosti modifikátorem.

## Podélné tahy



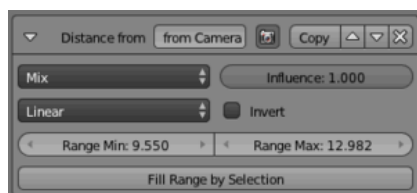
Modifikátor alfa pro podélné tahy

Modifikátor podélného tahu - Along Stroke mění hodnotu alfa v průběhu lineárním, nebo uživatelsky definovaným.

Mapování - Mapping

Lineární přírůstek (od **0.0** do **1.0**), který může být invertován pomocí volby Invert, nebo upraven uživatelskou křivkou mapování.

## Vzdálenost od kamery - Distance from Camera



Modifikátor průhlednosti čáry v souvislosti se vzdáleností od kamery

Modifikátor vzdálenost od kamery mění hodnotu alfa lineárně v závislosti na vzdálenosti od kamery, nebo pomocí upravené křivky, kde vzdálenost je parametr.

Mapování - Mapping

Lineární přírůstek (od **0.0** do **1.0**), který může být invertován pomocí volby Invert, nebo upraven uživatelskou křivkou mapování.

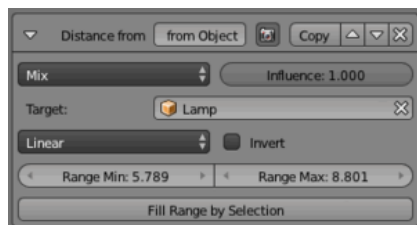
Rozsahy Range Min a Range Max

Meze mapování ze "vzdálenosti fotoaparátu" na "alfa v mapování". Pokud je aktuální bod vykreslení blíže, nežli Range Min, bude doplněna startovní hodnota alfa. A naopak je-li větší, než Range Max, bude brát nastavenou koncovou hodnotu alfa. Hodnoty vzdálenosti jsou v jednotkách ve scéně, nikoli pixelech!

Vyplnění rozsahu podle výběru

Pomůcka pro nastavení rozsahů (Fill Range by Selection) vzdáleností min/max od kamery na základě výběru objektů.

## Vzdálenost od objektu - Distance from Object



Vzdálenost od objektu

Modifikátor Vzdálenost od objektu - Distance from Object mění alfa stejným způsobem podle vzdálenosti od definovaného objektu.

### Cíl - Target

Objekt pro měření vzdálenost "odkud".

### Mapování - Mapping

Lineární přírůstek (od **0.0** do **1.0**), který může být invertován pomocí volby Invert, nebo upraven uživatelskou křivkou mapování.

### Rozsahy Range Min a Range Max

Meze mapování ze "vzdálenosti od objektu" na "alfa v mapování". Pokud je aktuální bod vykreslení blíže k cíli, nežli Range Min, bude doplněna startovní hodnota alfa. A naopak je-li vzdálenost od cíle větší než Range Max, bude doplněna nastavená koncovou hodnotu alfa. Hodnoty vzdáleností jsou v jednotkách ve scéně, nikoli pixelech!

### Vyplnění rozsahu podle výběru

Pomůcka pro nastavení rozsahů (Fill Range by Selection) vzdáleností min/max od objektu na základě výběru požadovaných.

## Materiál



Materiálový alfa modifikátor

Modifikátor materiálu doplňuje alfa podle aktuálního materiálu nacházejícím se pod tahem.

Je možné využít různé vlastnosti materiálů, z nichž některé jsou více komponentní(tj. vrací hodnoty RGB). V tomto případě bude použita průměrná hodnota.

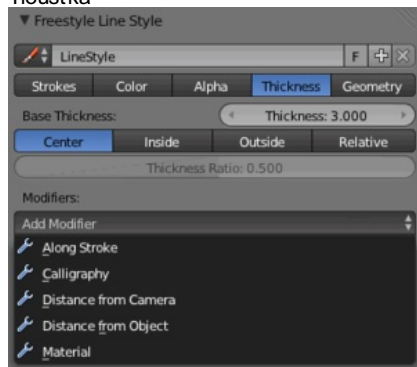
### Mapování - Mapping

Lineární přírůstek (od **0.0** do **1.0**), který může být invertován pomocí volby Invert, nebo může být upraven uživatelskou křivkou.

Jako poznámku je vhodné doplnit, že lineární neinvertovaná volba je ekvivalentem pro "neudělat nic", tj. původní hodnoty z materiálu jsou stále v rozsahu [0.0, 1.0]...

Pokud je použit s prvkem oddělení podle materiálu (Split by Material) v záložce Stroke, výsledek podélných tahů nebude mezi materiály rozmazaný.

## Tloušťka



### Rozhraní tloušťky čáry

V této záložce je možné řídit tloušťku čáry, tahu.

### Základní tloušťka

Základní tloušťka čáry pro daný styl.

### Pozice tloušťky

Ovládání polohy tloušťky tahu z původní (backbone) geometrie. K dispozici jsou čtyři možnosti:

#### Střed - Center

Tloušťka je rovnoměrně rozdělena na levou a pravou stranu geometrie tahu.

#### Uvnitř - Inside

Tahy jsou kresleny uvnitř hranice objektu.

#### Vně - Outside

Tahy jsou kresleny vně hranice objektu.

#### Relativní - Relative

Umožní specifikovat relativní pozici číslem v rozsahu mezi **0.0** (uvnitř) a **1.0** (vně) v numerickém poli poměr tloušťky (Thickness Ratio) o kousek níže.

Možnosti tloušťky pozice jsou použitelné na hranové tahy typu Silhouette a Border, protože to jsou okrajové typy definované z hlediska objektu hranice. Tahy jiných okrajových typů jsou vždy nakresleny pomocí možnosti Center.

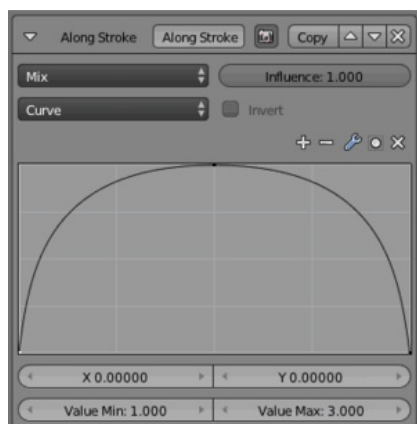
## Modifikátory

Je k dispozici pět modifikátorů tloušťky, které mohou být míchány se základní tloušťkou pomocí sady obvyklých metod ( viz např. [Mix kompozitní uzly](#)). Stejně jako u jiných modifikátorů Blender zpracovává modifikátory zásobníku shora dolů.

### Vliv

Vliv (influence) říká, jak mnoho bude mít modifikátor vliv na výsledný efekt tloušťky kresby.

## Podélné tahy



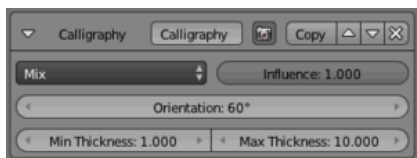
### Modifikátor tloušťky podélných tahů

Modifikátor Podélného tahu - Along Stroke upravuje základní tloušťku lineární, nebo uživatelskou funkcí podle délky tahu.

### Mapování

Mapování (mapping) lineární závislosti (od **0.0** do **1.0**), která může být invertována (volbou Invert), nebo uživatelskou funkcí.

## Kaligrafie



### Modifikátor tloušťky kaligrafické čáry

The Calligraphy modifier mimics some broad and flat pens for calligraphy. It generates different thickness based on the orientation of the stroke.

#### Orientation

The angle (orientation) of the virtual drawing tool, from the vertical axis of the picture. For example, an angle of **0.0°** mimics a pen aligned with the vertical axis, hence the thickest strokes will be the vertical ones, and the thinnest, the horizontal ones.

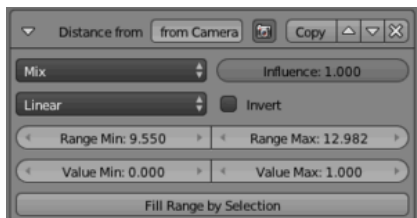
#### Min Thickness and Max Thickness

The minimum and maximum generated thickness (as explained above, minimum is used when the stroke's direction is perpendicular to the main Orientation, and maximum, when aligned with it).



Demonstrace modifikátoru kaligrafie - od T.K. [File:Toycar Calligraphy.zip](#)

### Vzdálenost od kamery



### Modifikátor tloušťky tahu na základě vzdálenosti od kamery

Modifikátor vzdálenost od kamery (Distance from Camera) mění tloušťku čáry lineárně, nebo podle definované křivky v závislosti na vzdálenosti kamery.

#### Mapování

Mapování (mapping) lineární závislosti (od **0.0** do **1.0**), která může být invertována (volbou Invert), nebo uživatelskou funkcí.

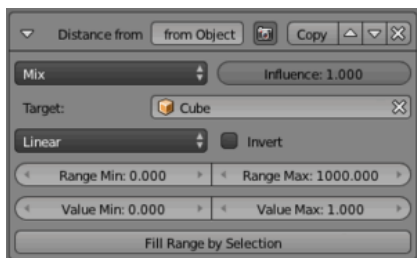
#### Range Min and Range Max

The limits of the mapping from “distance to camera” to “thickness in mapping”. If the current point of the stroke is at Range Min or less from the active camera, it will take the start thickness of the mapping, and conversely, if it is at Range Max or more from the camera, it will take the end thickness of the mapping. These values are in the current scene's units, not in pixels!

#### Fill Range by Selection

Set the min/max range values from the distances between the current selected objects and the camera.

### Vzdálenost od objektu



## Line Style Thickness's Distance from Object modifier

The Distance from Object modifier alters the base thickness with a new one from either a linear progression or a custom curve, using the distance to a given object as parameter.

### Target

The object to measure distance from.

### Mapování

Mapování (mapping) lineární závislosti (od **0.0** do **1.0**), která může být invertována (volbou Invert), nebo uživatelskou funkcí.

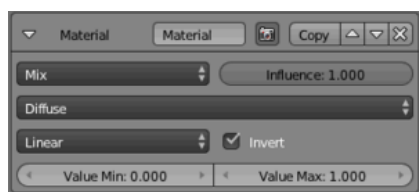
### Range Min and Range Max

The limits of the mapping from “distance to object” to “alpha in mapping”. If the current point of the stroke is at Range Min or less from the target, it will take the start thickness of the mapping, and conversely, if it is at Range Max or more from the target, it will take the end thickness of the mapping. These values are in the current scene's units, not in pixels!

### Fill Range by Selection

Set the min/max range values from the distances between the current selected objects and the target.

## Materiál



### Modifikátor tloušťky čáry podle materiálu

Modifikátor Materiálul mění tloušťku čáry podle materiálu, na který je čára kreslena.

Je možné využít různých vlastností materiálu, také vícekomponentní výsledky RGB. V tomto případě je využita hlavní hodnota.

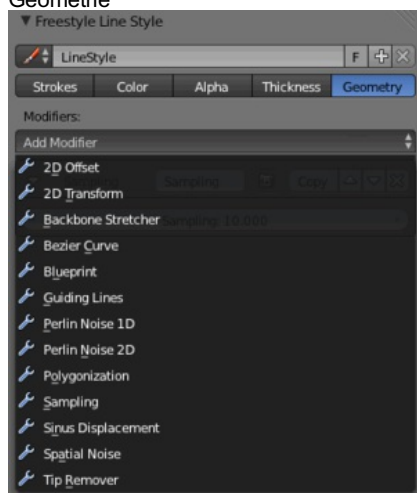
### Mapování

Mapování (mapping) lineární závislosti (od **0.0** do **1.0**), která může být invertována (volbou Invert), nebo uživatelskou funkcí.

Lieární neinvertovaná funkce znamená "nedělej nic", a vrací v mapování vstupní hodnoty v rozsahu [0.0, 1.0]...

Je-li použita volba rozdělení materiálem (Split by Material) v záložce tahu Stroke, výsledek nebude podél jednotlivých tahů rozmazaný mezi materiály.

## Geometrie



Celkové nastavení geometrie

V této záložce se řídí geometrie jednotlivých tahů.

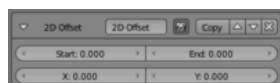
## Modifikátory

Je dostupných celkem třináct geometrických modifikátorů. Mají za úkol upravovat, deformovat dvojrozměrné freestyle tahy.

Tak jako je v Blenderu obvyklé, modifikátory jsou uplatňovány v zásobníku odzhora dolů.

### 2D Posuv

Tento 2D Offset - posuv modifikátor přidává jednoduchý posuv linek. Má dvě skupiny nezávislých voleb / efektů:



Modifikátor posuvu čar "2D Offset"

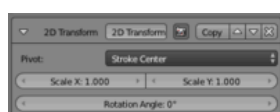
#### Start a End-konec

Tyto dvě volby určují velikost posuvu u začátku a konce čáry podél 2D normály. Výsledek je vytvoření "obalové křivky". Pokud například nastavíme Start na hodnotu **50**, bude počátek tahu posunut 50 pixelů, střed čáry bude 25 pixel. a konec tahu nebude posunut vůbec.

#### X a Y

Tyto dvě volby jednoduše přidají horizontální a vertikální posuv celému tahu.

### Transformace 2D



Line Style Geometry's 2D Transform modifier

The 2D Transform modifier applies two-dimensional scaling and/or rotation to the stroke backbone geometry. Scale is applied before rotation.

The center (pivot point) of these 2D transformations can be:

#### Stroke Center

The median point of the stroke.

#### Stroke Start

The beginning point of the stroke.

#### Stroke End

The end point of the stroke.

#### Stroke Point Parameter

The Stroke Point Parameter factor controls where along the stroke the pivot point is (**0.0** means start point; **1.0** end point).

#### Absolute 2D Point

The Pivot X and Pivot Y allows you to define the position of the pivot point in the final render (from the bottom left corner).

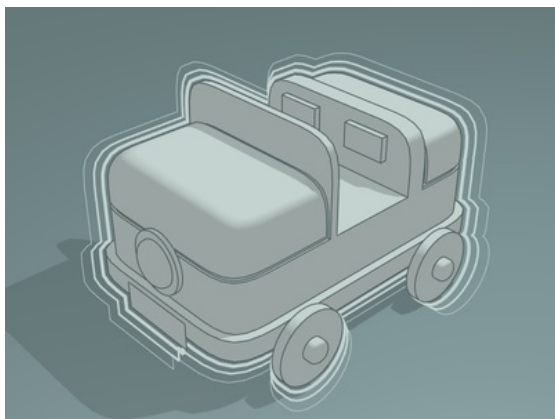
**WARNING:** Currently, you have to take into account the *real* render size, i.e. resolution **and** resolution percentage!

#### Scale X and Scale Y

The scaling factors, in their respective axes.

#### Rotation Angle

The rotation angle.



2D Transform modifier [File:ToyCar Three Contours.zip](#)

## Backbone Stretcher

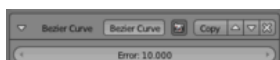


Line Style Geometry's  
Backbone Stretcher modifier

The Backbone Stretcher modifier stretches (adds some length to) the beginning and end of the stroke.

**Backbone Length**  
Length to add to the strokes' ends.

## Bezier Curve



Line Style Geometry's Bezier  
Curve modifier

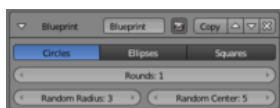
The Bezier Curve modifier replaces the stroke by a Bezier approximation of it.

**Error**  
The maximum distance allowed between the new Bezier curve and the original stroke.



Bezier Curve modifier demo by T.K. [File:ToyCar bezier.zip](#)

## Blueprint



Line Style Geometry's  
Blueprint modifier

The Blueprint modifier produces blueprint-like strokes using either circular, elliptical, or square contours. A blueprint here refers to those lines drawn at the beginning of free-hand drawing to capture the silhouette of objects with a simple shape such as circles, ellipses and squares.

Shape

Which base shapes to use for this blueprint: Circles, Ellipses or Squares.

#### Rounds

How many rounds are generated, as if the pen draws the same stroke several times (i.e. how many times the process is repeated).

#### Random Radius and Random Center

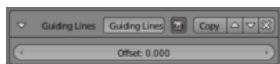
For the Circles and Ellipses shapes. Adds some randomness to each round in the relevant aspect. Using more than one round with no randomness would be meaningless, as they would draw over each other exactly.

#### Backbone Length and Random Backbone

For the Squares shapes. The first adds some extra length to each edge of the generated squares (also affected by the second parameter). The second adds some randomness to the squares.

Note that the Min 2D Length feature from the Strokes settings is quite handy here, to avoid the noise generated by small strokes...

### Guiding Lines



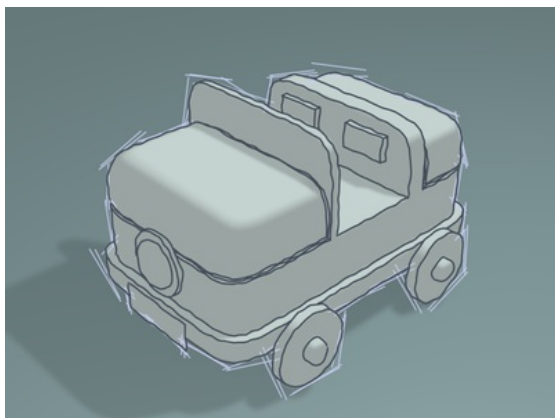
Line Style Geometry's  
Guiding Lines modifier

The Guiding Lines modifier replaces a stroke by a straight line connecting both of its ends.

#### Offset

Offset the start and end points along the original stroke, before generating the new straight one.

This modifier will produce reasonable results when strokes are short enough, because shorter strokes are more likely to be well approximated by straight lines. Therefore, it is recommended to use this modifier together with one of the splitting options (by 2D angle or by 2D length) from the Strokes panel.



Guiding Lines modifier Demo by T.K. [File:ToyCar Guiding Line.zip](#)

### Perlin Noise 1D



Line Style Geometry's Perlin  
Noise 1D modifier

The Perlin Noise 1D modifier adds one-dimensional Perlin noise to the stroke.

#### Frequency

How dense the noise is (kind of a scale factor along the stroke).

#### Amplitude

How much the noise distorts the stroke in the Angle direction.

#### Seed

The seed of the random generator (the same seed over a stroke will always give the same result).

#### Octaves

The "level of detail" of the noise.

#### Angle

In which direction the noise is applied (**0.0°** is fully horizontal).

### Perlin Noise 2D



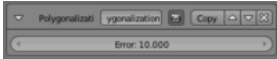
Line Style Geometry's Perlin Noise 2D modifier

The Perlin Noise 2D modifier adds one-dimensional Perlin noise to the stroke.

Its settings are exactly the same as the Perlin Noise 1D modifier.

TODO: What's the difference between those two modifiers?

## Polygonization



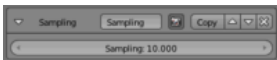
Line Style Geometry's Polygonization modifier

The Polygonization modifier simplifies strokes as much as possible (in other words, it transforms smooth strokes into jagged polylines).

Error

The maximum distance allowed between the new simplified stroke and the original one (the larger this value is, the more jagged/approximated the resulting polylines are).

## Sampling



Line Style Geometry's Sampling modifier

The Sampling modifier changes the definition, precision of the stroke, for the following modifiers.

Sampling

The smaller this value, the more precise are the strokes. Be careful; too small values will require a huge amount of time and memory during render!

## Sinus Displacement



Line Style Geometry's Sinus Displacement modifier

The Sinus Displacement modifier adds a sinusoidal displacement to the stroke.

Wavelength

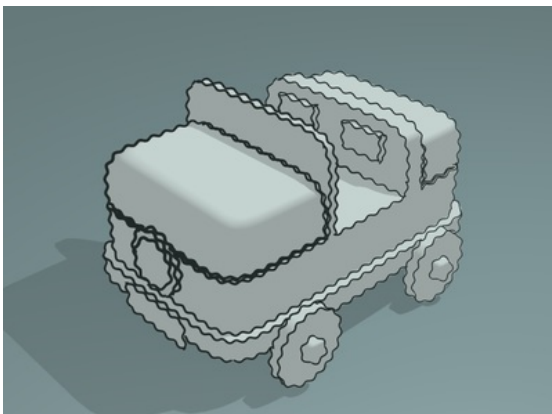
How wide the undulations are along the stroke.

Amplitude

How high the undulations are across the stroke.

Phase

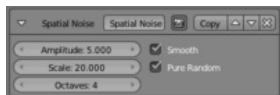
Allows "offsetting" ("moving") the undulations along the stroke.



Sinus Displacement modifier demo by T.K. [File:Toycar](#)

[Sinus.zip](#)

## Spatial Noise



Line Style Geometry's  
Spatial Noise modifier

The Spatial Noise modifier adds some spatial noise to the stroke.

TODO: definition of "spatial noise"!

### Amplitude

How much the noise distorts the stroke.

### Scale

How wide the noise is along the stroke.

### Octaves

The level of detail of the noise.

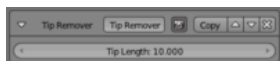
### Smooth

When enabled, apply some smoothing over the generated noise.

### Pure Random

When disabled, the next generated random value depends on the previous one; otherwise they are completely independent. Disabling this setting gives a more "consistent" noise along a stroke.

## Tip Remover



Line Style Geometry's Tip  
Remover modifier

The Tip Remover modifier removes a piece of the stroke at its beginning and end.

### Tip Length

Length of stroke to remove at both of its tips.

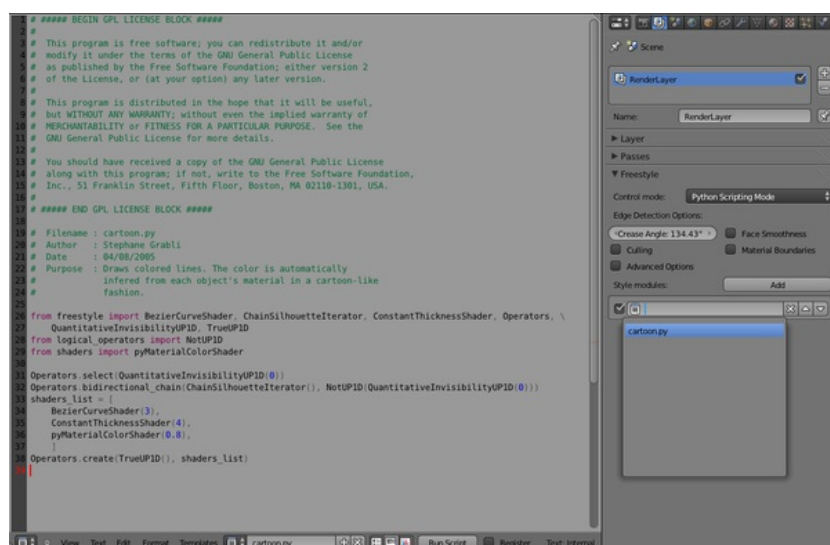
## skriptovací režim Python

Skriptování pythonem nabízí plnou programovatelnost a stylizaci kreslicích čar. V této kapitole jsou všechny operace zapsány do blender skriptu, který nakonec vystupuje v terminologii Freestyle jako modul. Vstupem je pohledová mapa (t.j. skupina detekovaných hran a charakteristik) a výstupem je sada stylizovaných tahů.

Stylový modul je složen z pěti základních operátorů: selection, chaining, splitting, sorting a stroke creation.

The selection operator identifies a subset of input feature edges based on one or more user-defined selection conditions (predicates). The selected edges are processed with the chaining, splitting and sorting operators to build chains of feature edges. These operators are also controlled by user-supplied predicates and functions in order to determine how to transform the feature edges into chains. Finally, the chains are transformed into stylized strokes by the stroke creation operator, which takes a list of user-defined stroke shaders.

Python style modules are stored within .blend files as text datablocks. External style module files first need to be loaded in the Text Editor window. Then the pull-down menu within an entry of the style module stack allows you to select a module from the list of loaded style modules.



A screen capture of a style module (cartoon.py) loaded in the Text Editor window (left), as well as Freestyle options in the Python Scripting mode in the Render Layers buttons (right)

Freestyle for Blender comes with a number of Python style modules that can serve as a starting point of your own style module writing. See also the section of the Freestyle Python API in the Blender Python API reference manual for the full detail of style module constructs.



By T.K. using the Python Scripting mode  
([File:Turning Pages.zip](#), CC0)



By T.K. using the Python Scripting mode  
([File:Lily Broken Topology.zip](#), CC0)

## Vytvoření modulu pro styl

A style module is a piece of code responsible for the stylization of Freestyle line drawing. The input of a style module is a set of feature edges called view map (ViewMap). The output is a set of stylized lines also referred to as strokes. A style module is structured as a pipeline of operations that allow for building strokes from the input edges within the view map. There are five kinds of operations (corresponding operator functions in parentheses):

- Selection (Operators.select())
- Chaining (Operators.chain(), Operators.bidirectional\_chain())
- Splitting (Operators.sequential\_split(), Operators.recursive\_split())
- Sorting (Operators.sort())
- Stroke creation (Operators.create())

The input view map is populated with a set of ViewEdge objects. The selection operation is used to pick up ViewEdges of interest to artists based on user-defined selection conditions (predicates). Chaining operations take the subset of ViewEdges and build Chains by concatenating ViewEdges according to user-defined predicates and functions. The Chains can be further refined by splitting them into smaller pieces (e.g., at points where edges make an acute turn) and selecting a fraction of them (e.g., to keep only those longer than a length threshold). The sorting operation is used to arrange the stacking order of chains to draw one line on top of another. The chains are finally transformed into stylized strokes by the stroke creation operation applying a series of stroke shaders to individual

chains.

ViewEdges, Chains and Strokes are generically referred to as one-dimensional (1D) elements. A 1D element is a polyline that is a series of connected straight lines. Vertices of 1D elements are called 0D elements in general.

All the operators act on a set of active 1D elements. The initial active set is the set of ViewEdges in the input view map. The active set is updated by the operators.

## Výběr

The selection operator goes through every element of the active set and keeps only the ones satisfying a certain predicate. The `Operators.select()` method takes as the argument a unary predicate that works on any `Interface1D` that represents a 1D element. For example:

```
Operators.select(QuantitativeInvisibilityUP1D(0))
```

This selection operation uses the `QuantitativeInvisibilityUP1D` predicate to select only the visible `ViewEdge` (more precisely, those whose quantitative invisibility is equal to 0). The selection operator is intended to selectively apply the style to a fraction of the active 1D elements.

It is noted that `QuantitativeInvisibilityUP1D` is a class implementing the predicate that tests line visibility, and the `Operators.select()` method takes an instance of the predicate class as argument. The testing of the predicate for a given 1D element is actually done by calling the predicate instance, that is, by invoking the `__call__` method of the predicate class. In other words, the `Operators.select()` method takes as argument a functor which in turn takes an `Interface0D` object as argument. The Freestyle Python API employs functors extensively to implement predicates, as well as functions.

## Chaining-řetězení

The chaining operators act on the set of active `ViewEdge` objects and determine the topology of the future strokes. The idea is to implement an iterator to traverse the `ViewMap` graph by marching along `ViewEdges`. The iterator defines a chaining rule that determines the next `ViewEdge` to follow at a given vertex (see `ViewEdgeIterator`). Several such iterators are provided as part of the Freestyle Python API (see `ChainPredicateIterator` and `ChainSilhouetteIterator`). Custom iterators can be defined by inheriting the `ViewEdgeIterator` class. The chaining operator also takes as argument a `UnaryPredicate` working on `Interface1D` as a stopping criterion. The chaining stops when the iterator has reached a `ViewEdge` satisfying this predicate during the march along the graph.

Chaining can be either unidirectional (`Operators::chain()`) or bidirectional (`Operators::bidirectional_chain()`). In the latter case, the chaining will propagate in the two directions from the starting edge.

The following is a code example of bidirectional chaining:

```
Operators.bidirectional_chain(ChainSilhouetteIterator(),
                             NotUP1D(QuantitativeInvisibilityUP1D(0)))
```

The chaining operator uses the `ChainSilhouetteIterator` as the chaining rule and stops chaining as soon as the iterator has come to an invisible `ViewEdge`.

The chaining operators process the set of active `ViewEdge` objects in order. The active `ViewEdges` can be previously sorted using the `Operators::sort()` method (see below). It starts a chain with the first `ViewEdge` of the active set. All `ViewEdges` that have already been involved in the chaining process are marked (in the case of the example above, the time stamp of each `ViewEdge` is modified by default), in order not to process the same `ViewEdge` twice. Once the chaining reaches a `ViewEdge` that satisfies the stopping predicate, the chain is terminated. Then a new chain is started from the first unmarked `ViewEdge` in the active set. This operation is repeated until the last unmarked `ViewEdge` of the active set was processed. At the end of the chaining operation, the active set is set to the Chains that have just been constructed.

## Rozdělení

The splitting operation is used to refine the topology of each Chain. Splitting is performed either sequentially or recursively. Sequential splitting (`Operators::sequentialSplit()`) in its basic form, parses the Chain at a given arbitrary resolution and evaluates a unary predicate (working on 0D elements) at each point along the Chain. Every time the predicate is satisfied, the chain is split into two chains. At the end of the sequential split operation, the active set of chains is set to the new chains.

```
Operators.sequentialSplit(TrueUP0D(), 2)
```

In this example, the chain is split every 2 units. A more elaborated version uses two predicates instead of one: One to determine the starting point of the new chain and the other to determine its ending point. This second version can lead to a set of Chains that are disjoint or that overlap if the two predicates are different. (see `Operators::sequentialSplit()` for more details).

Recursive splitting (`Operators::recursiveSplit()`) evaluates a function on the 0D elements along the Chain at a given resolution and find the point that gives the maximum value for the function. The Chain is then split into two at that point. This process is recursively repeated on each of the two new Chains, until the input Chain satisfies a user-specified stopping condition.

```
func = Curvature2DAngleF0D()
Operators.recursive_split(func, NotUP1D(HigherLengthUP1D(5)), 5)
```

In the code example above, the Chains are recursively split at points of the highest 2D curvature. The curvature is evaluated at points along the Chain at a resolution of 5 units. Chains shorter than 5 units won't be split anymore.

## Třídění

The sorting operator (`Operators::sort()`) arranges the stacking order of active 1D elements. It takes as argument a binary predicate used as a "smaller than" operator to order two 1D elements.

```
Operators.sort (Length2DBP1D())
```

In this code example, the sorting uses the Length2DBP1D binary predicate to sort the Interface1D objects in the ascending order in terms of 2D length.

The sorting is particularly useful when combined with causal density. Indeed, the causal density evaluates the density of the resulting image as it is modified. If we wish to use such a tool to decide to remove strokes whenever the local density is too high, it is important to control the order in which the strokes are drawn. In this case, we would use the sorting operator to insure that the most "important" lines are drawn first.

## Vytvoření tahů

Finally, the stroke creation operator (`Operators::create()`) takes the active set of Chains as input and build Strokes. The operator takes two arguments. The first is a unary predicate that works on Interface1D that is designed to make a last selection on the set of chains. A Chain that doesn't satisfy the condition won't lead to a Stroke. The second input is a list of Shaders that will be responsible for the shading of each built stroke.

```
shaders_list = [
    SamplingShader(5.0),
    ConstantThicknessShader(2),
    ConstantColorShader(0.2,0.2,0.2,1),
]
Operators.create(DensityUP1D(8,0.1, IntegrationType.MEAN), shaders_list)
```

In this example, the DensityUP1D predicate is used to remove all Chains whose mean density is higher than 0.1. Each chain is transformed into a stroke by resampling it so as to have a point every 5 units and assigning to it a constant thickness of 2 units and a dark gray constant color.

## User control on the pipeline definition

Style module writing offers different types of user control, even though individual style modules have a fixed pipeline structure. One is the sequencing of different pipeline control structures, and another is through the definition of functor objects that are passed as argument all along the pipeline.

Different pipeline control structures can be defined by sequencing the selection, chaining, splitting, and sorting operations. The stroke creation is always the last operation that concludes a style module.

Predicates, functions, chaining iterators, and stroke shaders can be defined by inheriting base classes and overriding appropriate methods. See the reference manual entries of the following base classes for more information on the user-scriptable constructs.

- UnaryPredicate0D
- UnaryPredicate1D
- BinaryPredicate0D
- BinaryPredicate1D
- UnaryFunction0DDouble
- UnaryFunction0DEdgeNature
- UnaryFunction0DFloat
- UnaryFunction0DId
- UnaryFunction0DMaterial
- UnaryFunction0DUnsigned
- UnaryFunction0DVec2f
- UnaryFunction0DVec3f
- UnaryFunction0DVectorViewShape
- UnaryFunction0DViewShape
- UnaryFunction1DDouble
- UnaryFunction1DEdgeNature
- UnaryFunction1DFloat
- UnaryFunction1DUnsigned
- UnaryFunction1DVec2f
- UnaryFunction1DVec3f
- UnaryFunction1DVectorViewShape
- UnaryFunction1DVoid
- ViewEdgeIterator
- StrokeShader

Odkazy

Zde jsou uvedeny odkazy na některé užitečné externí informace (angličtina).

## Videa

[\[video link\]](#)  
[The Light At The End](#)

[\[video link\]](#)  
[mmd\\_tools test2 with Blender+Freestyle \(未来時計 AM4:30\)](#)

## Video úvodníky

[\[video link\]](#)  
[An introduction to Freestyle plugin for Blender : "sketching" Suzanne / HD](#)

[\[video link\]](#)  
[Using freestyle in blender](#)

[\[video link\]](#)  
[Tutorial: Blender 3D - Freestyle and Composite](#)

[\[video link\]](#)  
[Blender Tutorial: Freestyle](#)

## Úvody

Základy Freestyle

<http://studiollb.wordpress.com/2012/02/29/freestyle-introductory-tutorial/>  
<http://jikz.net/archives/364>  
<http://jikz.net/archives/329>

Typy hran

<https://studiollb.wordpress.com/2012/09/08/freestyle-101-edge-types/>

Základní styly čar

<http://studiollb.wordpress.com/2012/09/08/freestyle-101-line-style-basic/>

Modifikátory stylů čar

<http://studiollb.wordpress.com/2012/09/08/freestyle-101-line-style-modifier-part-1/>  
<http://studiollb.wordpress.com/2012/09/08/freestyle-101-line-style-modifier-part-2/>  
<http://studiollb.wordpress.com/2012/09/15/freestyle-101-planning-and-along-stroke-line-style-modifier/>

Tipy and triky

<http://studiollb.wordpress.com/2012/02/03/freestyle-tips/> (Old)

## Různé

- [FreeStyle Users' improvement suggestions.](#)
- [FreeStyle integration into Blender blog](#)
- [Early documentation of FreeStyle](#)

## Kompozitní uzly

Kompozitní uzly je nástroj, který umožňují sestavit, upravit, modifikovat obraz respektive video. Pomocí uzlů je možné spojovat dva různé záběry v jeden, měnit charakteristiku barev statickým, nebo dynamickým způsobem. Práce s kompozitními uzly je intuitivní, podbízí k experimentům a dává uživatelům neuvěřitelné možnosti kombinací jednotlivých metod zpracování videa, či samotných obrázků. Jednotlivé uzly pracují jako *vstupné* - *výstupní jednotka*, která má za úkol určitým způsobem procházející data zpracovat.

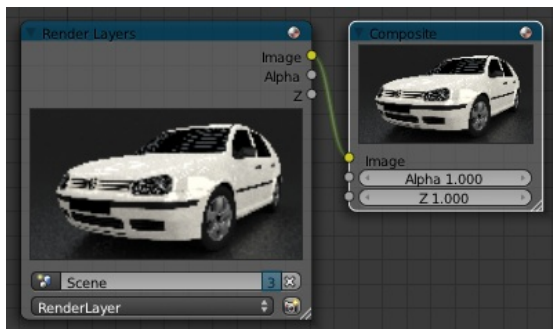
Term: Image

Používáme pojem *image* - *obrázek*, nebo také *snímek* pro jeden snímek v určitém časovém pořadí všech obrázků videosekvence. Pracovní *uzel* je funkční jednotka pro zpracování právě jednoho obrázku bez ohledu na typ vstupu.

Term: Noodle

Používáme pojem *nudle* - *noodle* pro soustavu propojení několika uzlů pomocí datových toků. Každá spojnice reprezentuje datový proud mezi uzly. Tím protékají obrazová, nebo řídicí data směrem z výstupu prvního uzlu do vstupu uzlu druhého. Z jednoho výstupu může vycházet více toků, avšak jako vstup do uzlu může být pouze tok jediný (pro spojování toků v jeden k tomu slouží speciální *spojovací uzel*).

*Pozn.překl.: pojem "nudle" může připadat čtenáři jako velice vtipný. Nicméně jinou alternativu, respektive přesné vyjádření pro "směs spojení" jsem nenašel. Pojem nudle je mnohem výstižnější, nežli například pojem "špagety", který se též v informatice často používá v souvislosti se složitostí propojováním heterogenních systémů.*



Chcete-li zpracovat soubory z disku, můžete pomocí uzlů importovat obrázek do Blenderu, upravit, případně sloučit s jinými obrázky a nakonec opět uložit na disk.

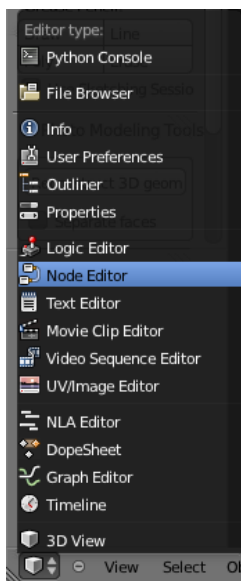
Příklad napravo znázorňuje nejjednodušší zapojení *nudlí*, kde vstupní uzel snímá tok z kamery a dále výstupní uzel, který tento proud dat ukládá.

## Přístup a aktivace uzlů

Přístup k [editoru uzlů](#) je možné povolením volby Composite Nodes - kompozitní uzly kliknutím na ikonu *obrázku*.

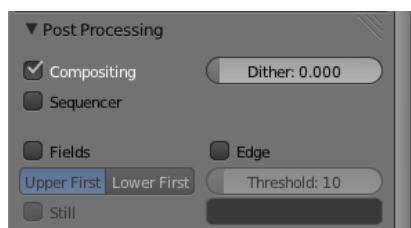


Editor uzlů Composite Nodes povolen



Vyberte okno editor uzlů

Pro aktivaci uzlů pro skládání zaškrtněte políčko Use Nodes - použití uzlů. Blender vytvoří výchozí počáteční soustavu nudlí, která se skládá ze dvou propojených uzlů.



Použití kompozice nudlí

Chcete-li použít tuto minimapu, musíte v Blenderu zapnout použití kompozitních uzlů mapy, která byla vytvořena a kompozitního obrazu pomocí složení uzlů. Přepněte tlačítkem Render a aktivujte tlačítko **Compositing - kompozitní** umístěné pod záložkou Post Processing. To říká Blenderu, aby komponoval výsledný obraz prostřednictvím kompozičního uzlu mapy.

Nyní máte k dispozici první sadu nudlí, vstupní vrstvu Renderu připojenou k výstupnímu uzlu. Sem můžete přidávat a připojovat mnoho [typů kompozitních uzlů](#) do celkové velké mapy. Geometrické umístění na ploše nemá na proces vliv, ale pouze logika propojení vstupů a výstupů jednotlivých uzlů. Uživatel je omezen pouze dostupnou výpočetní pamětí. *Pozn. překl.: Zde si dovoluji citovat automatického tlumočnicka anglického textu: "Ize komponovat, co hrdlo ráčí"*.

## Příklady

S obrázky je možné upravovat libovolně v rozsahu možností různých uzlů. Kompozitní zpracování videa se často používá v oblasti tzv. klíčování, tj. pohybující se objekt například natočený skutečný objekt před modrým pozadím je možné spojit s animovanou částí filmu tak, že modré pozadí je odfiltrováno, respektive přeměněno určitým uzlem v alfa kanál.

Je možné změnit celkovou "náladu" obrazu:

- Pro vytvoření "studeného" pocitu v obrázku je přidán modrý nádech.
- Chcete-li přidat historický nádech snímku, můžete obraz změkčit.
- Chcete-li vyjádřit nenávisť a frustraci, přidejte červený nádech nebo zvýšte červenou komponentu barev. Film "Sin City" je pravděpodobně extrémní ukázka této možnosti.
- Zarážející, důležitá okamžitá událost může být zaostřena s zvýšením kontrastu.
- Šťastný pocit - ano, uhádli jste - přidat žlutou (stejně části červené a zelené, žádná modrá), a tím se pohled projasní a "vysluní".
- Prach, nečistoty ve vzduchu a mraky v záběru zvýší reálnost.

[Quark66](#) 22:04, 9 October 2013 (CEST)

## Typy kompozitních uzlů

Následující odkazy jsou tříděny podle typů uzlů, které jsou seskupeny na základě podobnosti jejich funkcí:

- [Input \(vstup\)](#) - Přidání prvku do mapy uzlů, například obraz, hodnotu.
- [Output \(výstup\)](#) - Zobrazuje výsledek procesu jako malý obrázek.
- [Color \(barva\)](#) - Manipuluje s barvami obrazu.
- [Vector \(vektor\)](#) - Manipuluje s intenzitami a reflexemi v obraze.
- [Filters \(filtry\)](#) - Proces vylepšení obrazu na základě výpočtu sousedících pixelů.
- [Convertors \(konvertory\)](#) - Separuje obraz do komponent, nebo konvertuje formáty.
- [Mattes \(masky\)](#) - Vytváří masky pro odmaskování oblastí v obraze..
- [Distortion \(zkřivení\)](#) - Mění tvar obrazu.
- [Groups \(skupiny\)](#) - Uživatelsky definované skupiny uzlů.

[Quark66](#) 22:14, 9 October 2013 (CEST)

## Editace sekvencí

Jako doplněk k modelovacím a animačním nástrojům obsahuje Blender plnohodnotný editor videosekvencí (Video Sequence Editor - VSE) stejně jako node-based editor, kterým je možné zpracovávat videstream.. [Kompozitní nody](#) pracují stejně dobře nad obrázky, nebo videostreamy.

Blender VSE pracuje na konceptuálně vysoké úrovni z pohledu produkčního procesu. Zpracovává videoútržky v čase jako kousky mozaiky. Během typického a velice komplexního pracovního postupu Blenderu je zapojeno několik postupů:

- Modelování a vytváření objektů
- Přiřazení materiálů a zavedení osvětlení a barev k objektům
- Základní animace, "rozpohybování" objektů
- Renderování vrstev pomocí kamer
- Použití kompozitních nodů k:
  - Vylepšení obrazu pomocí nastavení barev, přidání do scény speciálních efektů
  - Umístění obrazů do kompozice, střih
- Skladba videostřihů dohromady použitím VSE.

VSE v Blenderu je kompletní videoprodukční systém, který umožňuje kombinovat více videokanálu a efektů. Jeho funkcionalita je již v Blenderu od prvopočátku, dokonce i v dobách, kdy Blender neměl rozhodně tolik funkcí tak byla možnost produkovat kvalitní videosnímky. A pomocí dalších pluginů je možné získat téměř neomezené možnosti tvorby videoefektů.

Při použití VSE jsou nataženy jednotlivé videoklipy od začátku do konce, v některých případech je možné je překrýt. Dále jsou vkládány "stnívačky", přerody obrazu mezi videoklipy. Nakonec je synchronizována audiostopa. Výsledkem zpracování editorem VSE je celková videosekvence.

### Podpora FFMPEG

Podpora exportního formátu **avi/quicktime** pomocí kompresního mechanismu FFMPEG funguje (od ver.2.44) pouze v prostředí Linux a Windows. S touto podporou FFMPEG je možné spolu s videem ukládat audiostopy do jednoho výsledného souboru.

## Rozšíření Blenderu

Narozdil od mnohadelších programů není Blender monolitický, nebo statický. Jeho funkcionalitu je možné rozšiřovat pomocí [skriptování Pythonem](#) bez nutnosti měnit zdrojový kód Blenderu, nebo nutné rekompile programu.

## Addons

Addons uímožňují získat rozšířené funkce Blenderu. Je možné je zapínat jednotlivě pomocí voleb jednotlivých rozšiřujících modulů v okně uživatelského nastavení.

Outside of the Blender executable, there are literally hundreds of addons written by many people:

- Officially supported addons are bundled with Blender.
- Other **Testing** addons are included in development builds but not official releases, many of them work reliably and are very useful but are not ensured to be stable for release.

An Overview of all addons is available in this wiki in the [Scripts Catalog](#) and in the [Extensions tracker](#).

## Skripty

Apart from addons there are also scripts you can use to extend Blenders functionality:

- Modules: Utility libraries for import into other scripts.
- Presets: Settings for Blender's tools and key configurations.
- Startup: These files are imported when starting Blender. They define most of Blender's UI, as well as some additional core operators.
- Custom scripts: In contrast to addons they are typically intended for one-time execution via the [text editor](#)

## Ukládání vlastních skriptů

### Umístění souborů

All scripts are loaded from the `scripts` folder of the [local, system and user paths](#).

You can setup an additional search path for scripts in [User preferences](#) (User Preferences → File Paths).

### Instalace

Addons are conveniently installed through Blender in the User Preferences → Addons window. Click the Install from File... button and select the `.py` or `.zip` file.

To manually install scripts or addons place them in the `addons`, `modules`, `presets` or `startup` directory according to their type. See the description above.

You can also run scripts by loading them in the [text editor](#) window.